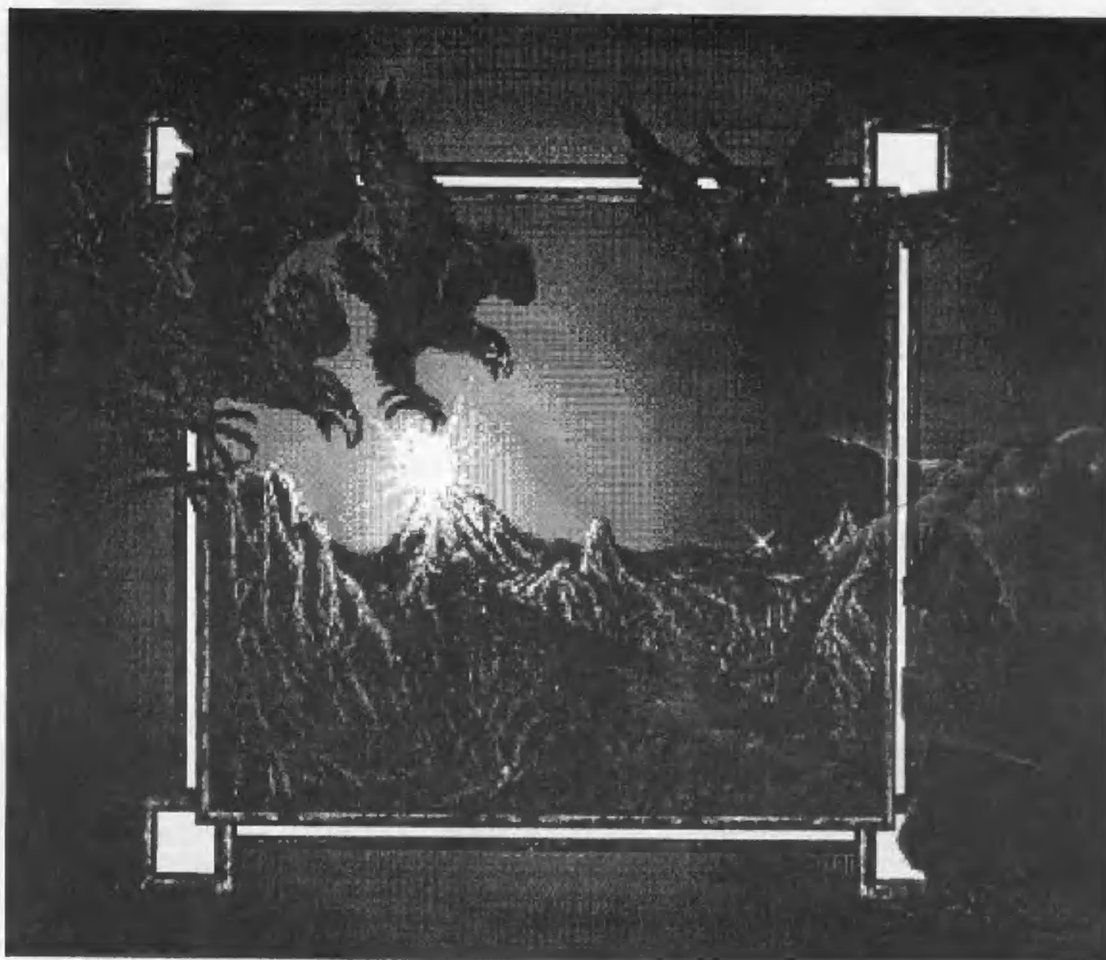
AMIGOWIEC

2/1992 Pismo użytkowników AMIGI cena 10.000 zł

Konkurs!



Spis treści



Kończy się zima, wielu z Was wróciło po zasłużonym odpoczynku do domów. Dla tych, którzy ciągle jeszcze tęsknią za górskimi krajobrazami przygotowaliśmy 'specjalną' okładkę. Nie są to wprawdzie ośnieżone Tatry, czy Beskidy, ale surowe góry Komputerlandu. Mamy nadzieję, że się Wam spodobają...

A co dzisiaj?
Już drugi dysk Public Domain. Nasza akcja ładowania na dyski tego co ciekawe, użyteczne, czy chociażby ładne spotkała się z dużym oddźwiękiem wśród naszych Czytelników. Mamy już sporo programików w zapasie, mamy też propozycje wydania, poza kolejnością, całodyskowego dema. Wszystko to oczywiście zależy od tego, czy spodobają się Wam nasze dyski. Dla pełniejszego obrazu świata dysków Public Domain i dla uniknięcia miana monopolistów nawiązaliśmy współpracę z grupą (?) nazywającą siebie "*Mysz Shareware*". Pierwszy opis tego co można znaleźć na dyskach PD (a jest tego oooooogromna masa) już za miesiąc.

Tak jak obiecaliśmy opisujemy jeden ze skanerów dostępnych do Amigi. Niestety większość z Was będzie rozczarowana brakiem możliwości wczytywania tekstów. Pamiętajcie jednak, że nie jest to wcale takie proste, że komputerowi daleko do zdolności rozpoznawania kształtów jakie ma czteroletnie dziecko. Jest to też pewna szansa dla Was (słowa te kieruję w tym momencie do tych, którzy programują) - rozwiązanie problemu, jak zmusić komputer do czytania, jest na pewno ciekawym wyzwaniem rzuconym człowiekowi przez bezmyślną maszynę.

W obecnym numerze opisujemy też wyniki testu następnego rodzimego produktu, jakim jest program o nazwie ***Skarabeusz***. Program ten jest na pewno godny, by znaleźć się w Waszej bibliotece programów. Pamiętajcie jednak, że nikt nie będzie pisał Wam programów za darmo i jeśli będziecie preferować 'kopie' od kolegi, to tym samym podetniecie gałąź na której siedzą nasi rodzimi programiści.

W tym miejscu apelujemy do wszystkich, którzy napisali jakiś program i chcieliby z nim zapoznać naszych Czytelników. Każdy przysłany do nas program, który będzie robił coś więcej niż dodawanie 2 do 2, a już na pewno każdy program komercyjny zostanie przez nas przetestowany i opisany. Jednocześnie podamy gdzie go można kupić (jest to jak najbardziej forma reklamy, z tym, że bezpłatna). Jeśli uważacie, że być może będziecie mieć jakieś zastrzeżenia do naszej recenzji - napiszcie nam o tym, a my odpowiednio wcześniej prześlemy Wam tekst recenzji, tak abyście mogli na niego odpowiedzieć - my zaś zamieścimy Waszą odpowiedź wraz z recenzją.

UWAGA: programy przesłane do redakcji, nie będą bez Waszej zgody w żaden sposób rozpowszechniane!

W obecnym numerze ogłaszamy też wspólnie z firmą TSS konkurs. Jego zasady są niesamowicie proste:
- wystarczy odpowiedzieć na postawione w nim pytania (str. 12). Wśród tych, którzy nadeślą prawidłową odpowiedź rozlosujemy ufundowaną przez TSS paczkę dyskietek. Na nadsyłanie odpowiedzi macie niecały miesiąc - zwycięzców zaś podamy za 2 miesiące w numerze majowym.

Czekamy na następnych sponsorów!

Przypominamy też o naszym konkursie na najciekawszy program - jeśli Twój program zamieścimy w Amigowcu (kod źródłowy), czy choćby na naszym dysku PD - otrzymasz za niego honorarium. Wprawdzie najniższe wynosi 20 tysięcy złotych, ale można otrzymać i kilkaset za program większy i bardziej dopracowany! Programu nie zamieścimy nigdzie i nie opublikujemy też jego kodu źródłowego bez Twojej wyraźnej zgody!

Aby zadość było sprawom formalnym - to oznajmiamy, że pamiętamy o naszej obietnicy rozdania 30 dyskietek z programami PD wśród pierwszych prenumeratorów i ich listę ogłosimy w następnym numerze.

Czekamy też na sygnały od Was, co chcielibyście widzieć na naszych łamach, jakie macie problemy i co Was trapi.

Tomasz Kokoszczyński

AMIGOWIEC
Luty 1992 cena 10.000 zł

Skład redakcji:
Redaktor naczelny
Tomasz Kokoszczyński
Zastępca redaktora naczelnego
Ryszard Kowalski
Kolegium redakcyjne:
Dariusz Zwierzyński (*grafika*)
Tomasz Flanc (*Public Domain*)
Jacek Ryć (*gry*)
Wojciech Białkowski,
Łukasz Bieńkowski
Tomasz Kulbacki
Krzysztof Nowicki
Maciej Klimkiewicz
Krzysztof Wirszyło
Jarosław Zachwieja

Wydawca:
ALFIN, Tomasz Kokoszczyński

konto: PKO I Bydgoszcz,
9511-64839-136

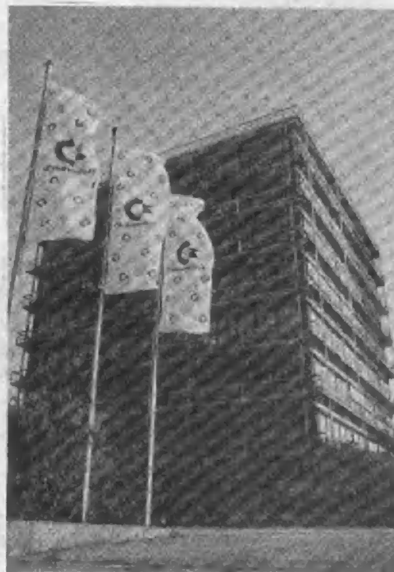
Adres Redakcji:
Ryszard Kowalski,
ul. Kasztanowa 50,
85-605 BYDGOSZCZ



COMMODORE wkracza do Polski!

Firma Commodore, znana wszystkim amigowcom z jednego ze swych najlepszych produktów, jakim jest Amiga, na stałe zadomawia się w naszym krajobrazie. 30 stycznia 1992 roku w hotelu Marriott odbyło się oficjalne otwarcie polskiego przedstawicielstwa tej firmy, kierowanej przez pana Andrzeja Drączkowskiego. Na otwarciu obecny był sam szef europejskiego oddziału firmy - *Helmut Jost.* Zapowiedział nawiązanie współpracy zarówno z naszymi rodzimymi handlowcami (przykładem jest tu firma JTT) jak i producentami rodzimego oprogramowania (no bo czymże jest najlepszy nawet komputer, bez duszy - czyli programu). Warto w tym miejscu zauważyć, że firma Commodore, jest nie tylko producentem Amig, CDTV, czy też starych, dobrych C-64. Oprócz wymienionych tutaj pozycji w asortymencie firmy znajdują się liczne IBMy - od 286 do 486. Jako ciekawostkę można podać, że w ostatnim czasie wielu wybitnych specjalistów z firmy IBM przeszło właśnie do Commodore... Trudno się więc dziwić, że firma zajmuje obecnie drugie, zaraz po IBMie, miejsce w produkcji i sprzedaży tych komputerów. Wracając jednak do Amig - pan Jost zapowiedział, że firma będzie dążyła do wdrożenia właśnie tego komputera w naszym szkolnictwie. Miejmy nadzieję, że to się uda i ta wspaniała maszynka zawita również do naszych szkół. Oczywiście jest to wielka szansa dla naszych rodzimych programistów - już dziś zapraszamy Was do współpracy - zarówno z firmą Commodore jak i z nami. Musicie bowiem wiedzieć, że i AMIGOWIEC nawiązał kontakty z Commodore i mamy nadzieję, że już w niedługim czasie będziemy w stanie jeszcze skuteczniej promować rodzimych twórców oprogramowania dla Amigi. Jak na razie efektem tej współpracy jest ostatnia strona okładki, ale zapewniamy Was, że na tym nie koniec!

Podsumowując wypada się cieszyć, że to właśnie w Polsce, a nie gdzie indziej otwarto wschodnio-europejskie przedstawicielstwo naszej ulubionej firmy. Na zdjęciu poniżej możecie zobaczyć jej frankfurcką siedzibę.



Amiga Laptop

Amerykańska firma *Never Technology* opracowała Amigę w wersji *Laptop.* Nie są jeszcze znane szczegóły, poza tym, że dwa Laptopy, roboczo zwane *Model 10* i *Model 30*, są nieco większe niż kartka formatu A4 i mają około 5 cm. grubości. Jako monitor służy 10-calowy ekran LCD w kolorze lub monochromatyczny.

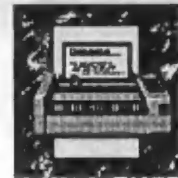
Model 10 posiada procesor 68000 taktowany 7,14 MHz i Kickstart 1.3 lub 2.0, zaś model 30 ma już Motorolę 68030 z zegarem 24 MHz i koprocesor matematyczny 68882. Obie wersje mają po 2 MB RAM na płycie (rozszerzalne do 8 MB) i standardową stację dysków 3,5" 880 KB. W fazie projektowania znajdują się urządzenia peryferyjne, takie jak np.wewnętrzny twardy dysk i karta PC. Nie wiadomo jeszcze, czy i kiedy te komputery wejdą do seryjnej produkcji. Poważne kłopoty producentom może sprawić fakt, że wszystkie Custom-Chipy Amigi są strzeżone prawami patentowymi (w przeciwieństwie do IBM PC XT i AT - dlatego pecety składa się u nas po garażach i piwnicach), a bez tych kości nie można zbudować funkcjonującego klonu Amigi. Precedensu w takiej sytuacji dostarczyła firma *Outbond,* która skonstruowała trzy odmiany notebooka kompatybilne z MacIntosh`em i uruchomiła produkcję bazującą na kościach uzyskanych ze skupu starych komputerów. Pozytywnym skutkiem ubocznym tej operacji było wypuszczenie przez Apple własnego, tańszego laptopa, jako reakcji na posunięcie konkurencji.

Niezbędnik programisty.

Książki z serii *Amiga ROM Reference Manul* pisane przez specjalistów firmy Commodore są podstawowym źródłem informacji dla wszystkich, którzy chcieliby dogłębnie poznać swoją Amigę. Obecnie ukazało się zaktualizowane wydanie książki z tej serii z podtytułem ***DEVICES.*** Autorzy omawiają i pokazują na przykładach jak używać systemowych interfejsów urządzeń AMIGI. Książka w pełni dostosowana jest do wersji 2.0. Można tu znaleźć opisy takich urządzeń jak: Console Device, Clipboard, Timer, Track Disk i wiele innych. Jakby na dodatek dołożono jeszcze pełny opis struktur IFF. Nie muszę tu chyba dodawać, że książka została napisana w uniwersalnym języku ludzi komputerów, czyli po angielsku.
Wydana przez wydawnictwo Addison -Wesley kosztuje 28.95 funtów.

Robot w szkole.

System Alfreda Robota współdziałający z Amigą jest przeznaczony do nauki dzieci i młodzieży podstawowych zasad robotyki. Niskie koszty systemu (?) powodują, że idealnie nadaje się do szkół, techników, czy nawet uniwersytetów. System składa się z robota, który może wykonywać proste czynności (czytaj: ruchy) oraz specjalnego pakietu programów. Uczący się mogą bezpośrednio na ekranie obserwować ułożenie poszczególnych elementów robota. Zasada działania zbliżona jest do zasady działania wielkich robotów przemysłowych.



3D-Pro v2.0

Bardzo popularna na Zachodzie karta do obróbki obrazu o wdzięcznej nazwie *Video Blender* została ostatnio wyposażona w nową wersję programu 3D-Professional służącego do tworzenia trójwymiarowych animacji (ray-tracing). Wersja 2.0 programu oferuje: łatwe konstruowanie brył i ich wypełnianie, operacje na obiektach w czasie rzeczywistym, obróbkę i konstrukcje brył trójwymiarowych.

Rozszerzenia, rozszerzenia.

ProRAM 3000 to nazwa nowego rozszerzenia pamięci do Amigi 3000. Pozwala rozszerzyć komputer od 4 do 64 MB co jest bardzo potrzebne w poważnej pracy z grafiką trójwymiarową.

Jak grzyby po deszczu powstają nowe karty 24-bitowej grafiki.

RAMBRANT najnowsza karta graficzna przeznaczona dla Amig 2000 i 3000. Jest wyposażona w podwójny bufor obrazu i daje maksymalną rozdzielczość 1024x1024 w 16.7-milionach kolorów. Na karcie jest zamontowany procesor graficzny z serii RISC taktowany zegarem 40 MHz i z dołączanym opcjonalnie kooprocesorem 34082.

DMI RESOLVER firmy *Digital Micronics* daje rozdzielczość 1280x1024. Posiada wbudowany 16-bitowy procesor graficzny Texas Instruments (RISC) taktowany zegarem 60 MHz. Wyświetla na ekranie 256 kolorów z palety 16 milionów. Bardzo dużo programów graficznych współpracuje z tą kartą.

VAN GOGH VIDEO ENHACER CARD tak brzmi pełna nazwa karty zwiększającej rozdzielczość naszego komputera. Daje ona cztery różne tryby od *VGA* na 1024x768 (*SVGA*) skończywszy. Karta wymaga 1 MB pamięci chip, lecz bardziej preferuje Agnusa w wersji 2 MB. Lepiej współpracuje ze wszystkimi komputerami wyposażonymi w procesor 68030 i wyżej.

Picture collection.

Brain Technologies oferuje program *PIX LIB*. Jest to zbiór przekonwertowanych z innych komputerów rysunków i grafik. Baza zawiera ich 995 i kosztuje 9.95$.

Krajobraz na ekranie.

Pojawiły się nowe programy służące do generowania krajobrazów. Firma *Virtual Reality Laboratories* zaprezentowała ostatnio program *Distant Suns 4.0* oraz nową wersję programu *Vista Professional*. W odpowiedzi firma *MagageM* wypuściła uzupełnioną wersję programu o nazwie Fractal Pro. Jego wersja 5.0 pracuje ze wszystkimi daoychczasowymi wersjami Amigi, w pracy uwzględnia karty turbo i nowe procesory. Jedyną wadą tych programów jest pamięciożerność. Wszystkie działają powyżej 3MB pamięci.

2D->3D

Pixel 3D v2.0 to najnowszy produkt fimy *Axiom Software*. Program służy do konwersji rysunków dwuwymiarowych w trójwymiarowe. Zawiera nową funkcję *Beveling* pozwalającą na konwersję obrazków na różne formaty widziane przez programy animacji 3-D.

Nowy Ray na scenie.

Radiance Software wypuściła nowy program typu ray-tracing. Jego nazwa brzmi , *Raydance*. Firma twierdzi, że generuje on obrazki o niespotykanym dotąd realiźmie.

REAL 3D wraca.

Firma *Activa* oddała ostanio do testowania najnowszą wersję programu *Real 3D*. Beta wersja 1.4 zawiera pełne 24-bitowe wypełnienia. Cały zestaw został wzbogacony o program *Race Trace* pozwalający na konwersję zwykłych rysunków bitmapowych na wektorowe objekty.

City dla 3D.

Firma *Texture City* wypuściła zestaw objektów i wypełnień pod tą samą nazwą. Jest on przeznaczony dla użytkowników programów ray-tracing. Zawiera elementy miasta i wypełnienia w szesnastu milionach kolorów.

NewTek najdroższy?

Nie wiele produktów może być sprzedawanych na rynku z ceną powyżej 900$. Jednak oprogramowanie karty *Video Toaster* firmy *NewTek* może. Najnowsze jego egzemplarze są tego przykładem. Karta łącznie z oprogramowaniem w wersji 2.0 jest obecnie sprzedawana za 2495$. Na szczęście posiadacze starszego opogramowania karty, ponieważ to właśnie z softwarem związany jest wzrost ceny produktu, mogą wymienić je na nowe za mniej niż 395$. Jak twierdzą autorzy nowe programy o ponad 50% zwiększają ilość efektów karty.

Soft-Logic jest raz.

Hotlinks to program asystujący przy wymianie danych między innymi produktami Soft-Logica i Page Stream'em. Jest używany do pracy w sieci jako jeden z pakietu programów oferowanego przez firmę. Ponadto zestaw zawiera edytor tekstu o nazwie *PageLinner*, program do obróbki rysunków wektorowych *Art Expresion*, ponad 600 fontów wektorowych, 15 dysków wektorowych grafik, program graficzny *BME* pozwalający na dokładną obróbkę rysunków w różnych formatach oraz samego Page Stream'a 2.2.

Szybkość to GVP.

G-Force 040 to nazwa nowej karty turbo firmy GVP. Poprzez zastosowanie bardzo szybkiej 40 ns (tak, tak 40 nanosekundowej) pamięci i Motoroli 68040 taktowanej zegarem 28 Mhz firma GVP wysunęła się znowu na czoło jeżeli chodzi o szybkość. Karta daje osiągi 22 MIPS i 3.75 MFLOPS. Ponadto ma możliwość zwiększenia swoich wyników przez wymianę na 68040 taktowaną 33 Mhz, gdy tylko będzie dostępna. Karta umożliwia rozszerzenie Amigi 3000 do 24 MB w sumie.

Opracowali T.K. i R.K



Gdzie Bije Dzwon? cz.4

Wszystkich, którzy 'nie załapali się' na poprzednie odcinki cyklu "Gdzie Bije Dzwon?" informujemy, że cykl ten poświęcony jest temu jak używać komputera inaczej niż grając - czyli jak programować. Autor jednak nie omawia tu konkretnego języka programowania - musicie bowiem wiedzieć, że taki język to nie wszystko, ponieważ jak powiedział Bohdan w nr 0/91, wielu ludzi zna angielski, ale tylko jeden facet swego czasu napisał "Szekspira"... Krótko mówiąc: istnieje coś takiego jak *algorytmy* i przygotowanie programu. Samo programowanie to nie wszystko!

T.K.

Prywatny konkurs rozstrzygnięty!!!

Załamać się można ilością odpowiedzi (jest ich aż ... dwie), ale to najdobitniej świadczy o tym, że w polskich szkołach matematyki się nie uczy, tylko "naucza". Przypominam pytanie: jak policzyć szybciej niż komputer (w czas pracy komputera wliczamy czas konstruowania i wykonywania programu) sumę liczb od 1 do 1000000?

A oto algorytm działania (dla większej przejrzystości okrójmy nasz przedział sumowania do 1000):

1...Zauważmy, że pierwsza liczba z dodawanego zakresu (czyli jeden) dodana do liczby ostatniej (czyli 1000) daje nam liczbę 1001
2...Zauważmy, że druga liczba dodana do drugiej od końca daje ... także 1001
3...I ogólnie - n-ta liczba dodana do n-tej od końca da nam 1001
4...Jeśli zsumujemy dolną połowę zakresu z połową górną w wyżej pokazany sposób to otrzymamy 500 razy liczbę 1001. Po wymnożeniu 1001 przez 500 otrzymamy wyliczoną sumę, czyli 500500.

Znajomość tego algorytmu pozwala na obliczenie tego typu sum wielokrotnie szybciej, niż możliwe byłoby to przy pomocy komputera. I tu pytanie: czy wszystkie zadania warto rozwiązywać przy pomocy informatyki? Nie, nie mam na myśli zadań trywialnych (typu 2+2=...), których rozwiązanie na maszynie cyfrowej jest z zasady nieopłacalne (więcej się człowiek nameczy stukając w klawisze, niż jest to warte). Chodzi mi o złożone zadania, do których rozwiązywania został komputer stworzony. A więc ...



CZY KOMPUTER JEST DOBRY NA WSZYSTKO?

Aby odpowiedzieć sobie na to pytanie należy wyjaśnić parę rzeczy elementarnych. Po pierwsze, założyć należy, że wszystkie programy, które będziemy omawiać w dzisiejszych rozważaniach operować będą na liczbach zmiennoprzecinkowych (czyli komputerowej reprezentacji liczb rzeczywistych). Założenie to nie mija się w zasadzie z rzeczywistością, gdyż wszelkie programy realizujące jakieś metody numeryczne posługują się właśnie tym typem liczb. Po drugie, uświadomić sobie trzeba, czym są liczby rzeczywiste! Jest to NIESKOŃCZONY zbiór liczb, a więc taki, że między dowolnymi dwiema liczbami z tego zbioru można znaleźć kolejną liczbę należącą do tego zbioru. Uważni czytelnicy poprzednich części "DZWONU" powinni już zrozumieć, gdzie tkwi niebezpieczeństwo posługiwania się liczbami zmiennoprzecinkowymi. Mniej uważnym, lub zapominalskim, tłumaczę: Komputer z racji na ograniczoną ilość bitów, na których reprezentowana jest liczba nie może z nieskończoną dokładnością odwzorować wartości liczby w pamięci. Przykładowo: liczba pi (około 3.1416....), która z racji swej natury jest liczbą o nieskończonej ilości cyfr po przecinku nigdy (sic!) nie będzie dokładnie reprezentowana w pamięci maszyny cyfrowej. Jeden komputer umożliwi nam zapisanie jej z dokładnością do 10 cyfr po przecinku, drugi z dokładnością do 20, trzeci nawet z dokładnością do 100, ale zawsze liczby te będą obarczone *BŁĘDEM REPREZENTACJI*. I już na koniec - powiemy, że komputer ma *SILNĄ ARYTMETYKĘ*, jeśli umożliwia nam zapisanie liczby zmiennoprzecinkowej z dużą dokładnością.

SKĄD WIEDZIEĆ CO LICZYĆ?

Można to policzyć!!! Nie ma obawy, nie będę zanudzał Was stronami matematycznych wyprowadzeń - wstęp od redakcji do trzeciej części "DZWONU" spowodował moją niechęć do dalszych matematycznych eksperymentów. W zarysie podam natomiast to, co wiedzieć trzeba, aby mieć zdrowe podejście do komputera.

Pierwszym testem opłacalności opracowywania zadania na komputerze jest policzenie *WSKAŹNIKÓW UWARUNKOWANIA ZADANIA*. Wskaźniki te charakteryzują wpływ zaburzenia danych na zaburzenie rozwiązania. Innymi słowy, wskaźniki te mówią, czy (zakładając, że wprowadzane do komputera dane zmiennoprzecinkowe są reprezentowane z pewnym błędem) błędy danych nie wpłyną negatywnie na reprezentację wyniku. Śmieszną sprawą jest to, iż uzyskanie wyniku negatywnego (tzn. jeśli wskaźniki powiedzą nam, że zadanie jest bezsensowne) eliminuje możliwość rozwiązywania tego problemu na komputerze. Na nic się zdaje tu zmiana algorytmu obliczeń - jedyną radą na poprawę wskaźników jest użycie zadania rownoważnego, ale dobrze uwarunkowanego, lub ewentulanie zmiana arytmetyki na silniejszą (czyli przykładowo zmiana Amigi na CRAY'a).

Drugim testem jest policzenie *WSKAŹNIKÓW KUMULACJI ALGORYTMU*. Zaraz na początku widać, że w przeciwieństwie do wskaźników uwarunkowania, które charakteryzowały zadanie, wskaźniki kumulacji charakteryzują algorytm. Zmieniając algorytm, który charakteryzuje się złymi wskaźnikami kumulacji można próbować rozwiązać zadanie inną drogą. Dobre wskaźniki kumulacji posiadają *ALGORYTMY NUMERYCZNIE POPRAWNE*, czyli takie, które dają rozwiązanie będące nieco zaburzonym dokładnym rozwiązaniem zadania o nieco zaburzonych danych.

Z polskiego na nasze: zakładając, że dane zaburzone są na poziomie reprezentacji maszynowej (czyli są to zaburzenia wynikające tylko i wyłącznie z siły arytmetyki) otrzymamy wykonując algorytm numerycznie poprawny wynik także zaburzony na poziomie reprezentacji maszynowej.



Mówiąc jeszcze prościej - wskaźniki kumulacji określają, czy błędy, które ZAWSZE podczas obliczeń powstają zostają skumulowane w wyniku obliczeń, czy też zostaną po prostu w nim uwzględnione (ale bez ich kumulacji). Algorytmy użyteczne (sensowne do zaimplementowania na komputer) posiadają wskaźniki kumulacji rzędu ilości działań algorytmu (a więc widać, że błędy nie zostają w wyniku działań algorytmu wzmocnione).

Trzecim testem sensowności implementowania zadania jest sprawdzenie stabilności numerycznej algorytmu. Kiedy algorytm jest stabilny? Jeśli mały błąd na dowolnym etapie obliczeń zostaje przenoszony dalej z nie rosnącą amplitudą (czyli następuje redukcja lub przynajmniej nie zwiększanie się błędu na wszystkich etapach obliczeń). Nie jest na przykład algorytmem stabilnym algorytm liczenia bardzo małych wartości przy pomocy rekurencyjnej metody, która w pierwszym kroku przyjmuje za wynik algorytmu wartość bardzo dużą. Dlaczego? Bo bardzo duża wartość w kroku początkowym może mieć błąd reprezentacji maszynowej DUŻO WIĘKSZY od rzeczywistego wyniku zadania!!!

Jak widać, aby zabrać się do pisania poważniejszych programów trzeba nie tylko zaprojektować program z punktu widzenia informatyki i komputera, ale także i od strony matematycznej. Oczywiście - powiecie, że można tego nie robić i też będzie dobrze. W 95% przypadków będziecie mieli rację. Autorzy tych pozostałych 5% przekonają się natomiast na własnej skórze co to znaczy stracony czas (nierzadko parę miesięcy pisania programu...). Popatrzmy jednak na program jeszcze na chwilę z punktu widzenia maszyny. Zastanówmy się...

JAKI JEST KOSZT WYKONANIA PROGRAMU?

Oprócz jakości algorytmu ważny jest też jego koszt, czyli liczba działań arytmetycznych i logicznych, jaką musi wykonać komputer, aby obliczyć wynik. Nie jest to może tak ważne, gdy zajmujemy się pisaniem programów dla własnego, domowego użytku. Jednak szybkość działania programu, związana przecież z ilością wykonywanych działań, jest najczęściej pierwszą wnerwiającą nas rzeczą, jaką w programie zauważamy (ach, ten czas obliczeń w Sculpt'cie 4D...). I co z tego? I nic!!! Komputer jest od tego aby liczył - ALE - należy starać się konstruować takie algorytmy rozwiązywania zadań aby minimalizować liczbę działań (minimalizować koszt wykonywania programu). W rezultacie takich przemyśleń powstała nowa gałąź "wiedzy" zajmująca się ZŁOŻONOŚCIĄ OBLICZENIOWĄ algorytmu. Na jej bazie powstało parę stwierdzeń, które, z grubsza, mogą dać wyobrażenie o kosztach wykonania programu. Po pierwsze: udowodniono, że jeśli zadanie ma N istotnych danych to minimalna liczba działań potrzebnych do obliczenia wyniku wynosi N/2. Po drugie: dla wielu zadań można udowodnić, iż minimalna liczba działań w algorytmie numerycznie poprawnym musi być istotnie większa od liczby danych. A po trzecie: statystycznie rzecz biorąc istnieje bardzo mało optymalnych metod obliczania konkretnych zadań (mamy więc pole do popisu - można główkować i wymyślać optymalne algorytmy).

CO Z TEGO WSZYSTKIEGO WYNIKA?

Wynika z tego mnóstwo śmiesznych (może tylko dla laika, ale zawsze) rzeczy. Człowiek nie wiedzący czym jest komputer myśli przeważnie, że wystarczy sformułować pewien problem (czasami doda jeszcze trochę danych, ale nie zawsze uważa, że są one potrzebne) i po paru sekundach komputer poda gotową odpowiedź. Nic bardziej złudnego!!! Przede wszystkim - może tej odpowiedzi wcale nie podać (albo podać, ale będzie ona zła). Może ją podać, ale nie po paru sekundach, a po paru ... latach pracy!!! Może ją także podać i po paru sekundach, ale ... nie będziemy wiedzieć, czy podany wynik jest poprawny. Bez zdroworozsądkowego (lub matematycznego - jak kto woli) podejścia do problemu możemy po napisaniu programu, uruchomieniu i otrzymaniu wyniku wiedzieć na jego temat dokładnie tyle samo, ile wiemy przed rozpoczęciem prac nad programem. Wynik może być bowiem w takim stopniu zaburzony, że aż bezużyteczny. Warto więc mieć na uwadze i ten aspekt pisania i projektowania programu.

KOŃCZĄC JUŻ DZISIAJ...

... chciałbym podziękować śmiałkom za rozwiązanie problemu. Arkadiusz Pereguda oraz Marcin Ożdziński przesłali wyczerpujące i poprawne odpowiedzi na podane pytanie. I tu mam problem - nagroda jest jedna, a byłoby grzechem losować jednego zwycięzcę spośród dwóch uczestników! Nagroda zostanie więc podzielona - obydwaj uczestnicy dostaną po pięć dysków MAXELL MF-2DD oraz parę sztuk firmowych naklejek Commodore'a (poczynając od płachty metr na pół metra a kończąc na malutkich naklejeczkach). Nagrody wyślę oczywiście pocztą (znając polski cykl wydawniczy są już dawno na miejscu), a wszystkich, którzy plują sobie w brodę, że nie napisali do Amigowca w sprawie konkursu zapraszam do udziału w następnych. Obiecuję, że będzie jeszcze coś do rozwiązania.

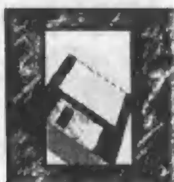
Do przeczytania w następnym odcinku, w którym zajmiemy się odwzorowaniem danych w komputerze, czyli strukturami danych.

Marek Hyla

Już w obecnym numerze udało się Markowi zrealizować tą obietnicę - jego firma sponsoruje konkurs, o którym możecie poczytać na stronie 12!

Zapraszamy!!!

Redakcja





Oto mam przed sobą kolejny produkt firmy Twin Spark Soft. Jest nim gra zatytułowana ***Skarabeusz***. Oryginalna (czytaj: angielska) nazwa tej gry brzmi - ***SCRABBLE*** i jest to gra bardzo popularna na Zachodzie - mamy nadzieję, że Skarabeusz zadomowi się i u nas, tym bardziej, że dzięki programowi, który dzisiaj opisujemy, można sobie potrenować w domu.

Zacznijmy może do samych zasad gry. Jak podaje instrukcja jest to gra planszowa, logiczno-słowna. Najkrócej mówiąc polega ona na tworzeniu własnych krzyżówek (jest jakby odwrotnością ich rozwiązywania). Krzyżówkę budujemy wspólnie z naszym przeciwnikiem na planszy o wymiarach 15*15 pól. Na planszy tej znajdują się kolorowe pola, które mają szczególne znaczenie - omówimy je za momencik.

Krzyżówkę tworzymy na przemian - jeden gracz jedno słowo, drugi nastepne i tak po kolei (w grze może uczestniczyć do 4 graczy).

Oczywiście w naszej grze tak jak i w normalnej krzyżówce słowa muszą do siebie pasować przynajmniej jedną literą. Słowa tworzymy z puli 7 liter, które mamy 'na ręku'. Również ograniczona jest pula wszystkich liter, które mamy do dyspozycji w ciągu całej gry - wynosi ona 100 'sztuk'.

Grę zaczynamy od centrum planszy. Pierwszy z graczy musi położyć swoje pierwsze słowo tak, aby jedna z liter, leżała na polu centralnym. Następni gracze kładą swoje słowa tak, aby chociaż jedna z liter pochodziła ze słowa leżącego już na planszy.

To praktycznie wszystko. Aha, spytacie pewnie co z tymi kolorami i za co można dostawać punkty. Otóż jedno wiąże się z drugim. Punkty dostajemy za poszczególne litery użyte do budowania naszych nowych słów. Każda litera 'z ręki' ma w dolnym prawym rogu liczbę punktów, które możemy za nią uzyskać. Sumuje się po prostu liczbę punktów i... No i te kolory. Kolory pól są 4 (oprócz szarego, który jest kolorem pola normalnego):

- pole jasnoczerwone podwaja nam sumę wszystkich punktów
- pole ciemnoczerwone potraja tą sumę
- pole jasnoniebieskie podwaja punkty leżącej na nim litery
- pole ciemnoniebieskie potraja punkty tej litery

Dodatkowo możemy uzyskać 50 punktów premii za wykorzystanie wszystkich liter 'z ręki'.

Jeśli dojdziemy do wniosku, że nie da się utworzyć, żadnego wyrazu z posiadanych liter to możemy je wymienić - kosztuje nas to jednak utratę ruchu.

Tyle gra, a co na to komputer?

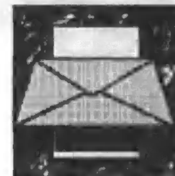
Jeśli kogoś zainteresował powyższy opis zasad gry SCRABBLE, czy też w spolszczonej wersji SKARABEUSZA, to powinien kupić sobie licencjonowaną wersję komputerową tej gry, wyprodukowaną przez firmę Twin Spark Soft.

Oryginalna gra, w odróżnieniu od pirackiej kopii, charakteryzuje się tym, że jest sprzedawana w estetycznym pudełku, na dyskietce z odpowiednią nalepką, wraz z instrukcją, która nie wygląda jak 6-te ksero - jest po prostu zupełnie normalną instrukcją.

Czym różni się komputerowa realizacja Skarabeusza od Scrabble planszowej?

Otóż normalnie gra się na planszy, literki zaś są rozrzucane po stole (grzbietami do góry) lub wyciągane z woreczka. Gracze sami ustalają, czy dane słowo istnieje, no i co najważniejsze musi być ich przynajmniej dwóch.

Gracz 1 114
Gracz 2 87
Gracz 2 : x--11 y-- 6 POZIOMO KIT 5



W Skarabeuszu komputerowym komputer załatwia wiele niewygodnych, formalnych spraw. On to 'stwarza' pulę liter, na bazie której odbędzie się gra, on to losowo przydziela po 7 literek, na każdą turę i wreszcie on to sprawdza, czy wyrazy są napisane prawidłowo. Komputer też może występować jako całkiem niezły przeciwnik - można zatem potrenować sobie w Skarabeusza nie mając nikogo pod ręką, można też urządzać turnieje w kilku graczy (jednym z nich lub kilkoma może być komputer).

Ma to jak widzimy ogromne zalety, ma też jednak i wady. Autorzy programu wyposażyli go w podstawowy zestaw słów, który składa się li tylko z rzeczowników (w liczbie pojedynczej i mnogiej) - a jak wiemy, w mowie polskiej występują także czasowniki, zaimki, przymiotniki itp. itd. Jednak na całe szczęście zestaw słów, dostępnych w komputerowym słowniku nie został ustalony raz na jutro i można go zmieniać. Uczenie komputera następuje przez dopisanie nowych słów. Nie popłaca tu oszukiwanie, ponieważ, w następnych rozgrywkach, komputer, będzie korzystał z wymyślonych przez nas słów, a nie zawsze będzie nam to na rękę... (a może by tak następna dyskietka ze słownikiem dla tych, którzy już mają Skarabeusza?).

Grę zaczynamy wkładając dyskietkę do stacji df0: i resetując komputer. Pojawia nam się czołówka z bardzo udanym podkładem muzycznym (aż nie chce się przechodzić dalej - podkład jest chyba za dobry).

Następnie przechodzimy do właściwej gry. Tu ustawiamy ilość graczy (do 4) i ustalamy, który z nich jest graczem komputerowym, a który człowiekiem. Przy graczach obsługiwanych przez komputer możemy ustalić siłę gry (im większa tym komputer dłużej będzie myślał). No i zaczynamy grać. Naprawdę nietrudno zorientować się w obsłudze (pamiętajmy, że niejasne kwestie zostały wyjaśnione w instrukcji). Dla początkowego zorientowania się jak to wygląda możemy sobie obejrzeć walkę komputera z komputerem (niestety, przy tej opcji nie udało mi się przełączyć podkładu muzycznego na odgłosy gry, nie za bardzo działał też klawisz ESC - Jest to na szczęście jedna z niewielu usterek jakie udało mi się znaleźć w tym programie).

Podsumowując: Skarabeusz jest całkiem udaną realizacją gry Scrabble i zastępuje z dużym powodzeniem papierowe plansze i literki. Gra ta wyposażona w słownik zapewnia, szczególnie bardzo młodym ludziom, zarówno rozszerzenie słownictwa jak i naukę prawidłowej pisowni (wyrazy nieprawidłowo zapisane, nie będą akceptowane). Skarabeusz uczy więc jak szybko, sprawnie i logicznie używać tego, co wyróżnia człowieka spośród gromady zwierząt - czyli mowy i to na dodatek jak najbardziej mowy polskiej. Gra ta ma charakter rozwojowy (rozwija się wraz z graczem) i jak to napisano w instrukcji: "Możliwość dopisywania słówek pozwoli dostosować poziom gry komputera do stanu Twojej wiedzy." My mamy jednak nadzieję, że za jakiś czas, autorzy programu poszerzą słownik, tak że gra ta stanie się jeszcze pełniejszym programem "rozwijającym zdolności słowotwórcze", co w kraju, gdzie najczęściej używanym przez ludzi bez wyobraźni słowem jest k...a, jest bardzo na czasie.

Program polecamy wszystkim, którzy chcieliby zerwać z tym schematem i nauczyć swoje pociechy prawidłowego posługiwania się tym pięknym narzędziem jakim jest poprawna polszczyzna oraz tym, którzy chcieliby spędzić wolny czas w gronie przyjaciół łącząc przyjemne z pożytecznym.

Skarabeusza przetestował

Tomasz Kokoszczyński

P.S. Program można nabyć bezpośrednio w firmie "*TWIN SPARK SOFT*" pisząc na jej adres:
Marek Hyla
os. Kolorowe 9/16
31-939 KRAKÓW
tel. (012) 44-43-68

TWIN TM SPARK SOFT.

Drodzy Czytelnicy!

Dzisiejsze odpowiedzi na Wasze listy będą miały nieco inny, 'krótszy' charakter. Mamy jednak nadzieję, że Wy nadal będziecie dopisywali i dostarczycie nam tematów do następnych rubryk pisma pisemka. Jednocześnie oczekujemy na Waszą pomoc w sprawach, z którymi sami nie możemy sobie dać rady (wspomnieliśmy o nich w poprzednim AMIGOWCU).

Aby uprzedzić wszelkie Wasze listy z pretensjami, że komuś nie posłaliśmy zamówionego egzemplarza, czy, że mamy jakieś opóźnienia informujemy, że bez podania adresu - a zdarzyło się to już kilka razy, nie będziemy w stanie odnaleźć zamawiającego. Poza tym musicie wziąć pod uwagę, że przekazy pieniężne idą do nas czasami i przez 3 tygodnie.

W styczniowym numerze "64 plus 4" zauważyłem alarmujące listy Czytelników o "Commodore Amiga Clubie" z Bełchatowa, o którym i my pisaliśmy w jednym z ostatnich wydań amigowego fanzinu. Piszą oni, że powysyłali tam dyskietki na zamówione programy i ślad po nich zaginął. Nie mam zamiaru roztrząsać w tym miejscu uczciwości ***Dariusza Chałady***, prezesa klubu, pragnę tylko zauważyć, że każdy sam musi rozstrzygnąć, czy stać go na zaryzykowanie kilku dyskietek dla zdobycia **kopii** poszukiwanych programów. Pamiętam jak sam ładnych kilka lat temu posłałem dyskietkę do niejakiego "***FanComClubu***" prowadzonego przez pana ***Marka Frąckowiaka*** z Poznania i tyle ją widziałem (a dodajmy tu, że wtedy dyskietka kosztowała 4 DM i było to o wiele więcej pieniędzy niż dzisiaj). Wniosek stąd zatem taki, że naiwnych nie sieją, a sami się rodzą, zaś każdy powinien uczyć się na własnych błędach. Niestety tak to już jest, że na pochyłe drzewo kozy skaczą, a na tym, że nie wszystkich stać na oryginalne programy korzystają ludzie bez skrupułów...

Rubrykę zredagował

Tomasz Kokoszczyński



AMIGOWY PUBLIC DOMAIN #2

Public Domain czyli "własność publiczna". Tym właśnie terminem zwykło określać się programy komputerowe, które nie stanowią niczyjej własności, a więc wszyscy użytkownicy mogą je posiadać i z nich korzystać...

W ten właśnie sposób rozpoczęliśmy, mniej więcej przed miesiącem, naszą nową akcję: sprzedaż dyskietek z programami powszechnie uznanymi za swobodnie rozpowszechnialne. Jak na razie, nasza inicjatywa spotkała się z dosyć dużym zainteresowaniem. Kilka osób udostępniło nam nawet swoje prace (niektóre z nich znajdują się już na naszym najnowszym, drugim dysku). Szczerze mówiąc, nie spodziewaliśmy się tego po tak krótkim okresie czasu jaki upłynął od wydania ostatniego numeru AMIGOWCA. Wszystkim zainteresowanym serdecznie dziękujemy.

I oto mamy drugi Amigowy Public Domain Pack. Większość stałych czytelników naszego czasopisma dobrze wie o co chodzi, ale domyślam się, że część z Was widzi AMIGOWCA po raz pierwszy. Dla nich właśnie należy się krótkie objaśnienie kilku najważniejszych pojęć ściśle związanych z programami udostępnionymi do publicznej prezentacji.

Public Domain

Programy bez zastrzeżonych praw autorskich. Można z nimi robić teoretycznie wszystko: kopiować, używać, adaptować, przerabiać wykorzystywać w innych programach.

Freeware

To programy, które można kopiować, ale mają zastrzeżone prawa autorskie. Wolno je używać i przegrywać (ale tylko w całości). Nie wolno ich sprzedawać, a jedynie kopiować za symboliczną opłatą (za kopiowanie).

Shareware

Czyli programy o ograniczonym prawie do ich używania. Prawo do legalnego posiadania programu uzyskać można dopiero po zarejestrowaniu się jako użytkownik. Następuje to, najczęściej, po zapłaceniu autorowi programu określonej sumy.

Demonstracyjne

Niepełne wersje programów, rozpowszechniane wyłącznie w celach reklamowych. Nie wszystkie z nich można kopiować bez ograniczeń.

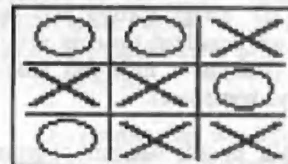
Tyle, w telegraficznym skrócie, na temat praw i nazewnictwa związanego z programami zamieszczanymi na naszych dyskietkach. Należy jednak dodać, że powyższy podział nie wyczerpuje wszystkich możliwości, zawierając tylko podstawowe grupy. Generalnie, wszystkie te programy łączy to, że mogą być swobodnie rozpowszechniane.

A CO NA DYSKIETCE #2?

Podobnie jak poprzednio, na dyskietce, oprócz kilku programów użytkowych, znajdują się również utwory muzyczne, prosta gra i animacja. Ale nie tylko. Dzięki uprzejmości jednego z bydgoskich koderów, lidera grupy Fluffy Bears, udało nam się stworzyć Mini Kącik Kodera, w którym co jakiś czas prezentować będziemy proste programy napisane w asemblerze. Do każdego z nich dołączony będzie program źródłowy wraz z opisem.

A oto lista tego, co znajduje się na dysku #2:

1. ***TicTacToe*** - to komputerowa wersja popularnej gry logicznej, znanej u nas pod nazwą "kółko i krzyżyk". Grę można wczytywać zarówno z poziomu WorkBencha, jak i CLI. To drugie rozwiązanie, choć bardziej pracochłonne, daje jednak parę ciekawych opcji. Można na przykład spowodować, że komputer będzie rozgrywał kolejne partie samoczynnie (wpisz: TicTacToe -0). Poza tym, po zakończeniu gry, wyświetlona zostanie lista wyników wszystkich rozegranych partii (zjawisko to nie występuje po uruchomieniu gry z poziomu WorkBencha).



2. ***CalcKey*** - to prosty, choć bardzo użyteczny programik. Przyda się on wszystkim tym, którzy wykorzystują komputer nie tylko do zabawy.

Program można nazwać emulatorem kalkulatora. Ale jest to kalkulator dość niezwykły. Przede wszystkim napisany jest w taki sposób, że można go w dowolnym momencie uaktywnić (np. podczas pracy z bazą danych lub arkuszem kalkulacyjnym). Po co? Chociażby po to, aby wprowadzić do bazy wynik kilku obliczeń, które musimy akurat wykonać. Tak, tak, wyprowadzanie wyniku obliczeń z kalkulatora do programu może odbywać się w sposób automatyczny.

Autor CalcKeya nie zapomniał również o programistach i wprowadził opcję wykonywania obliczeń w systemie heksadecymalnym, a także konwersji liczb systemu szesnastkowego na dziesiętny i odwrotnie.

Uaktywnienie programu umożliwia kombinacja klawiszy [Alt][=].

3. ***DosError*** - to program wyświetlający komentarz do podanego numeru błędu. Komentarz napisany jest oczywiście w języku angielskim.

Program pozwala na wyświetlenie jednego błędu lub pełnej listy. Po jego uruchomieniu należy podać numer błędu, do którego pragniemy zobaczyć komentarz lub też wpisać słowo ALL. Listę tę można również wydrukować na drukarce (DOSError >prt: ALL), albo zapisać jako plik tekstowy



(DOSError >df0:Nazwa ALL). Używając programu bezpośrednio z poziomu CLI, parametr podaje się zaraz po nazwie komendy, np: DOSError 121.
Rezultat wykonania tej komendy będzie następujący:
DOS Error 121: FILE NOT OBJECT
Co oznacza, że plik nie może zostać uruchomiony.

4. ***Lotto*** - "Miliard w środę, miliard w sobotę". Czy już wiesz o co chodzi? Lotto jest programem, który można nazwać symulatorem LOTTOMATU, czyli maszyny służącej do losowego wybierania liczb. Pomocny dla wszystkich miłośników gier liczbowych. Prosty w obsłudze. Wystarczy najechać kursorem na okienko z napisem LOTTO i nacisnąć lewy klawisz myszy. Spowoduje to, że program wygeneruje sześć liczb z zakresu od 1 do 49. Wciśnięcie lewego klawisza myszy podczas, gdy kursor znajduje się [redacted] okienku z napisem QUIT, powoduje wyjście [redacted] programu.



5. ***SysInfo*** - program dający pełną informację [redacted] systemie, który wykorzystujesz, czyli o komputerze i wszystkich urządzeniach do niego podłączonych. Otrzymujesz informację [redacted] ilości podłączonej pamięci, wersji KickStart'u, rodzaju specjalizowanych procesorów, napędów dyskowych, itp. Sprawdzenie wszystkiego trwa kilkanaście sekund. Proces testowania systemu następuje zaraz po uruchomieniu programu. Przejście do Workbencha możliwe jest wyłącznie po zakończeniu testu.

6. ***EgyptRemix*** - utwór muzyczny napisany przy pomocy programu NoiseTracker 2.0. Kompozycja dobrze [redacted] dawnym użytkownikom komputera Commodore 64. Przepisał na Amigę: BatBeast.

7. ***Fire, water, air*** - kolejny utwór muzyczny napisany [redacted] wykorzystaniem NoiseTrackera. Kompozycja autorstwa Marcina Gackowskiego pochodzi [redacted] pierwszego dysku muzycznego bydgoskiej grupy OLD BULLS.

8. ***Dioda*** - to pierwszy program w Mini Kąciku Kodera, napisany przez Stalkera. Efektem jego uruchomienia jest miganie diody zasilania w Amidze. Na dysku znajduje się również program źródłowy wraz z opisem, napisany w Seka Asemblerze.

9. ***Animacja*** - wykonana przez Wojciecha Białkowskiego przy pomocy programów Imagine i Scenery Animator.

Autorem większości ikonek zamieszczonych na dysku jest Dariusz Zwierzyński.

<u>Do wszystkich programów załączone są krótkie opisy w języku polskim !</u>

Prosimy o uwagi o naszej bibliotece. Piszcie do [redacted] jakich programów poszukujecie i jakie powinny się w niej znaleźć. Czekamy również [redacted] owoce Waszej pracy.

<u>UWAGA:</u> Na trzecim z kolei dysku PD znajdzie się rewelacyjny program Arriba, który dzięki prostej sztuczce ożywi Twój komputer! Szczegóły [redacted] miesiąc, w trzecim (marcowym) [redacted] Amigowca.

*

Jak zamawiać i ile to kosztuje?

Dyskietki PD można zamawiać przesyłając pieniądze [redacted] konto (blankiet [redacted] stronie 13) oraz:

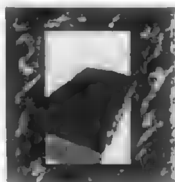
- podając numer dysku PD (na razie dostępny jest #1 i #2)
- podając CZYTELNIE swój adres
- podając ilość posiadanej pamięci w komputerze

Nasz dysk PD kosztuje 25 tysięcy złotych (włączone w to są koszty dysku, przesyłki, nalepki, opakowania, itp.).

<u>UWAGA:</u> Dla osób prenumerujących AMIGOWCA cena jest niższa i wynosi *<u>tylko</u>* 20 tysięcy złotych !

Jeżeli ktoś sobie życzy by mu wysłać dysk PD [redacted] zaliczeniem pocztowym, to [redacted] pokrywa koszty tego zaliczenia.

Opracował *Tomasz Hanc.*



Skaner

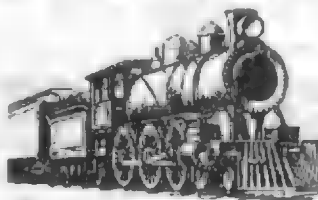
Skaner

Skaner

Smutna jest dola tzw. komputerów "domowych". Będąc potomkami elektronowych maszyn liczących, bardzo często znajdują zastosowanie nie mające nic wspólnego z obliczeniami (oczywiście mam [redacted] myśli obliczenia numeryczne, [redacted] nie te które dokonują się we wnętrzu każdego komputera). Weźmy [redacted] przykład taką Amigę: fantastyczny komputer o fascynujących możliwościach, a musi posłusznie wykonywać wszystkie rozkazy płynące z joystick'a i kierować czy to Lemmingami, czy to ogniem z broni maszynowej. Poza grami komputery najczęściej chyba wykorzystywane są jako inteligentne (i to bardzo) maszyny do pisania. Ale mimo tego określenia, trudno znaleźć edytor, który oferowałby tylko tyle, ile zwykła maszyna do pisania. Zdecydowana większość edytorów (nie mówiąc już o programach DTP) posiada możliwość włączenia do tekstu grafiki. Jednakże nawet najbardziej wyrafinowane możliwości importu grafiki nie zdadzą się na nic, jeśli nie [redacted] co importować. No, a skąd brać grafiki? Można korzystać z gotowych zestawów na dyskietkach (skąd brać dyskietki, chyba w naszym kraju wiadomo), ale co począć jeśli chcemy do dokumentu załączyć własne dzieło? Pozostaje skorzystać z usług programu graficznego (np. Deluxe Paint), bądź skanera.

Skaner to nazwa przyrządu służącego do przenoszenia rysunków, bądź zdjęć do komputera. Duży skaner formatu A4 (lub większego) przypomina niekiedy swoją konstrukcją kserokopiarkę, z tą różnicą, że nie wypluwa zadrukowanej kartki papieru, a tylko przesyła ciąg bitów do komputera. Wadą takiego urządzenia jest cena zaczynająca się od ok.1000 $. Bardzo często nie jest potrzebny rysunek tak dużego formatu; wtedy z powodzeniem można zastosować ręczny skaner. Wśród wielu dostępnych dla Amigi wybrałem skaner firmy *Golden Image* (pod taką nazwą występuje tajwańska firma *Jin Tech*) o symbolu *JS-105-1M* (1M oznacza amigową wersję, gdyż istnieją także wersje dla Atari ST i IBM PC).

Skaner ręczny (po angielsku *hand-held* lub *handy scanner*) jest to niewielkie urządzenie, które prze[redacted] się po kartce papieru, [redacted] ono posłusznie zamienia obraz na impulsy elektryczne zrozumiałe dla odpowiedniego programu komputerowego. Pod spodem skanera zamontowany jest szereg zielono-żółtych diod elektroluminescencyjnych (*LED*). Oświetlają one rysunek, [redacted] światło odbite o natężeniu zależnym od jasności danego punktu obrazka przechodzi przez szereg cieniutkich kanalików i pada na przetworniki *CCD* (*Charged Coupled Device*) - takie same, jak te w kamerach video. Reszta to zadanie dla przetwornika analogowo-cyfrowego i już do komputera wędruje obrazek w postaci cyfrowej.

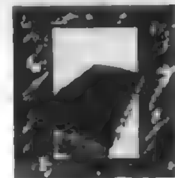


Po rozpakowaniu pudełka znajdujemy interface, zasilacz, dyskietkę z programem "Touch-Up", instrukcje, no i oczywiście [redacted] skaner. Skaner podłączany jest do Amigi przez mały interface instalowany do portu równoległego. Jest to bardzo dużym mankamentem, gdyż wielu użytkowników drukarek również wykorzystuje ten port, [redacted] interface wyposażony jest w bardzo małą wtyczkę, którą niezwykle trudno wyciągnąć bez użycia np. śrubokręta. Pewnym wyjściem może być odpowiedni przełącznik, ale przełączanie urządzeń przy włączonym komputerze może być czynnością ryzykowną. Ze względu na dość duży pobór prądu interface zasilany jest zewnętrznym zasilaczem zintegrowanym [redacted] wtyczką sieciową. Zarówno zasilacz, jak i skaner mają dosyć długie przewody, co znacznie ułatwia pracę.

Do pracy niezbędny jest program "*Touch-Up*", produkt amerykańskiej firmy *Migraph Inc.*, określany w instrukcji jako 'Program do obróbki czarno-białych obrazków wysokiej rozdzielczości'. I to zgadza się z prawdą, gdyż jest to program przede wszystkim graficzny, [redacted] tylko jedną z jego opcji jest obsługa skanera. Mimo tego, nawet po pierwszym uruchomieniu programu łatwo wpaść na to, co robić. Jest oczywiście instrukcja, ładnie wydana 170-stronicowa książka, ale (w przypadku skanerów sprowadzanych z Niemiec) napisana w języku niemieckim, który jest chyba mniej popularny wśród użytkowników Amig niż angielski. O możliwościach tego programu niech świadczy fakt, że spośród tych 170 stron tylko 20 poświęcono obsłudze skanera, [redacted] resztę funkcjom graficznym. Co prawda "Touch-Up" nie należy do najszybszych programów, niemniej da się wytrzymać...

Większe zastrzeżenia można natomiast mieć do instalacji tego programu (nie tylko [redacted] twardym, ale i na zwykłym dysku, gdyż dostarczona jest tylko dyskietka instalacyjna, [redacted] której nie można uruchomić programu) - oprócz jego przeniesienia, konieczne jest również zainstalowanie czcionki '*tufont*' w katalogu FONTS: Na dyskietce znajduje się program Install_Fonts (który kopiuje czcionki), ale Touch-Up'a kopiuje się albo przez przesunięcie szuflady Touch-Up do innego okna (jest to metoda o tyle niewygodna, że w tej szufladzie znajduje się katalog Images zawierający obrazki o łącznej objętości ponad 200 KB, które wraz z samym Touch-Up'em - ok.350 KB - dość skutecznie potrafią zapełnić dyskietkę) albo skopiowanie plik po pliku. Dla mnie ideałem byłby program instalacyjny, kopiujący wszystkie potrzebne pliki do odpowiednich katalogów.

Po uruchomieniu programu należy ustawić przy pomocy przełączników skanera odpowiednią rozdzielczość (100, 200, 300 lub 400 dpi - dots per inch, czyli punktów [redacted] cal), tryb (możliwe: tekstowy, czyli



czarno-biały, bądź 3 fotograficzne uwzględniające stopnie szarości), a także kontrast. Program umożliwia zdefiniowanie rozmiaru skanowanego obszaru (cała szerokość skanera - 105 mm, albo tylko połowa oraz długość od 5 do 35 cm). Maksymalna wielkość skanowanego obrazka zależy od ilości dostępnej pamięci. Mając Amigę z 1MB pamięci, w rozdzielczości 400 dpi trudno osiągnąć coś więcej niż 5cio centymetrowej długości obrazek o maksymalnej szerokości (105 mm). Po dokonaniu wszelkich wyborów klikamy myszką w odpowiedni gadżet, zapala się czerwona dioda na obudowie skanera, szereg diod umieszczonych pod skanerem i można zaczynać. Należy ustawić skaner na początku rysunku (aktualnie skanowany fragment widać dzięki okienku znajdującemu się obok diod), nacisnąć przycisk 'START' i powoli przesuwać. Jeśli czerwona dioda zacznie migać, jest to znak, że należy zwolnić, natomiast jeśli zgaśnie, trzeba powtórzyć operację od początku. Prędkość skanowania nie jest duża (szczególnie w największej rozdzielczości) i trzeba trochę wprawy, aby utrzymać skaner wzdłuż linii prostej. Po osiągnięciu zadanej długości wszystkie diody zgasną, a na ekranie komputera pojawi się rysunek. Można go nagrać na dysk w całości, bądź wybrać tylko jego fragment (Clip).

Tu chciałbym obszerniej napisać o formatach plików, jakich używa Touch-Up. Poza standardowam IFF plik można zapisać jako IMG, PCX (format IBM-owskiego programu Paintbrush), TIFF, GIF, Degas, MacPaint (to Macintosh), PrintMaster (to rysunek dla programu o takiej nazwie) i EPSF (Encapsulated PostScript File). Ponieważ do Touch-Up'a można wczytać większość tych plików (z wyjątkiem GIF i EPSF), może on być wykorzystywany jako doskonałe narzędzie do konwersji formatów zapisu monochromatycznych grafik. Używając jeszcze np. programów CrossDos, Dos-2-Dos lub Mac-2-Dos można uzyskać dostęp do grafik z IBM'a czy też nawet Macintosha, jak również przygotować pliki dla tych komputerów. Jest to ogromny plus Touch-Up'a!

Wszystkie obrazki tworzone przez Touch-Up'a są naturalnie monochromatyczne, jednak przy użyciu drobnego triku można uzyskać kolory. Otóż program posiada bardzo przydatną funkcję - możliwość przetworzenia zeskanowanego, czarno-białego obrazka na szesnastokolorowy rysunek w formacie IFF (naturalnie te kolory to odcienie szarości). Tak otrzymany obrazek można potem "pokolorować" przy użyciu np. DeluxePaint'a i przy odrobinie wprawy uzyskać dzieło zbliżone do oryginału.



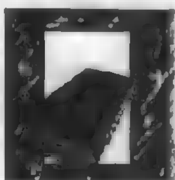
Specjalnie nie opisuję szczegółowo wszystkich funkcji programu, gdyż jak już wspomniałem, jego instrukcja ma 170 stron (czyli nawet jej skrót zająłby cały numer Amigowca), a jeśli ktoś kupi skaner, to i tak dostanie ten program wraz z instrukcją.

Niekiedy poprawek łatwiej można dokonać korzystając z np. Deluxe Paint'a (jest on nieco szybszy), jednakże mimo tego Touch-Up jest programem wartym zainteresowania ze względu na kilka unikalnych funkcji (np. wypełnianie dowolnego kształtu - Fill - przy użyciu jednego z zdefiniowanych wzorów, podobnie jak w Pagestream'ie czy też przekształcanie pełnej figury w sam kontur - funkcja Outline).

Zakup skanera pociąga za sobą naturalnie otrzymanie licencjonowanej kopii programu; wystarczy ją tylko zarejestrować (czyli wypełnić i wysłać kartę rejestracyjną). W moim przypadku firma Migraph Inc. zareagowała dośćszybko - po kilku tygodniach otrzymałem kilkustronnicową broszurkę pt."Headlines". Wynika z niej, że posiadana przeze mnie wersja 1.04 programu nie jest najbardziej aktualną; istnieją już 1.1 i 2.0 (tą pierwszą można otrzymać po opłaceniu tylko kosztów przesyłki, druga kosztuje 20$). Dodatkowo można zamówić program Scanning Utility, który umożliwia tylko nagranie skanowanego rysunku, a nie posiada możliwości jego edycji (przydatny, jeśli zaczyna brakować pamięci lub zamierzamy korzystać z innego programu graficznego). Jeśli ktoś zamierza skanować rysunki szersze niż 105 mm (szerokość skanera), to z pewnością ucieszy go możliwość zakupu programu Merge Utility, pomagającego łączyć rysunki. Firma także zapowiada stworzenie programu OCR (Optical Character Recognition), czyli narzędzia pozwalającego rozpoznawać literki w stworzonym przez skaner pliku graficznym poprzez porównanie ich z wzorcami zdefiniowanymi w programie. Co ciekawe ma on działać w oparciu o technikę Omnifont, czyli kształt poszczególnych liter zapamiętywany jest nie w postaci mapy bitowej, ale opisu matematycznego. Według zapewnień producenta pozwala to na przyspieszenie pracy programu, zwiększenie dokładności (do ponad 98%) oraz zmiejszenie zarówno jego rozmiarów, jak i wymagań pamięciowych.



Jeśli te zapowiedzi znalazłyby potwierdzenie w rzeczywistości, byłby to wielce interesujący program, zważywszy, że niektóre profesjonalne programy dla IBM PC (np. węgierska Recognita) w ogóle nie przewidują obsługi skanerów ręcznych, a jeśli zostanie wczytany do takiego programu plik przygotowany takim skanerem, to przeważnie jego odczytanie jest tak problematyczne, że bardziej opłaca się wstukać tekst na klawiaturze. Wszystkich niecierpliwie oczekujących na ten program muszę jednakże uprzedzić, że dobre wyniki daje odczytywanie tylko bardzo wyraźnie wydrukowanych tekstów. Tak więc nie powinno być większych problemów z czytelnym maszynopisem,



bądź wydrukiem [redacted] drukarki laserowej. Z drugiej strony wszelkie próby z wydrukiem z 9-cio igłowej drukarki [redacted] góry skazane [redacted] [redacted] niepowodzenie (litera złożona jest z oddzielnych punktów). Nie wiadomo również, czy program da się zmusić do rozpoznawania polskich liter...

Na koniec warto zwrócić uwagę [redacted] fakt, że [redacted] rynku znajduje się również kilka innych skanerów ręcznych. Do najczęściej spotykanych należą GeniScan firmy Datel Electronics oraz skanery firm Naksha i Pandaal. Co ciekawe, produkty firm Golden Image i Pandaal [redacted] to praktycznie te [redacted] skanery, różniące się tylko oprogramowaniem. Trudno mi porównywać te programy, ponieważ posiadam tylko skaner GI, niemniej [redacted] informacji zawartych w zachodnich czasopismach wynika, że Touch-Up, choć nieco wolniejszy, oferuje o wiele większą ilość funkcji. Typowy program graficzny dla cierpliwych... Z kolei GeniScan to też bardzo podobne urządzenie (w końcu trudno wymyślić inny [redacted] - ergonomia ma swoje wymogi) sprzedawane [redacted] z programem ScanEdit. Jest to mały programik oferujący kilka najprostszych operacji typu inwersja obrazka, odbicie lustrzane, czy też "postawienie go [redacted] głowie". Jak widać wszystkie skanery prezentują podobny poziom techniczny (ale także cenowy), wybór należy więc do użytkownika. Inna sprawa, że nie spotkałem dotąd [redacted] naszym rynku innego skanera niż Golden Image...

Podsumowując, skaner Golden Image mogę polecić wszystkim, którzy od czasu do czasu potrzebują wprowadzić do swoich dokumentów niewielkie obrazki, [redacted] nie opłaca im się kupować stacjonarnych urządzeń. Duża prostota obsługi, spore możliwości programu graficznego i dobra jakość [redacted] dokładność uzyskiwanych rysunków dodatkowo uzasadniają taki wybór.

Łukasz Bieńkowski

P.S. Oprócz skanera firma Golden Image produkuje wiele innych urządzeń peryferyjnych dla Amig (i nie tylko) - od myszek po rozszerzenia pamięci i dodatkowe stacje dysków. Te pierwsze polecam szczególnie wszystkim tym, którym dał się we znaki firmowy zwierzak Commodore'a (a szczególnie jego klawisze). Myszki GI używam od ponad pół roku i niezwykle wysoko cenię sobie jej precyzję [redacted] absolutną niezawodność klawiszy działających w oparciu [redacted] mikrowyłączniki (a nie membranowe blaszki).

TWIN TM SPARK SOFT.

KONKURS! TSS



Lutowy AMIGOWIEC został w znacznym stopniu opanowany przez Marka Hylę i firmę, której jest głównym 'działaczem', czyli Twin Spark Soft. Marek napisał następną część cyklu dla myślących pt." **Gdzie Bije Dzwon?**". On też jest jednym z autorów programu "**Skarabeusz**", którego recenzję zamieszczamy [redacted] stronie 6. No i nareszcie jego firma, czyli Twin Spark Soft ufundowała paczkę dysków dla jednej z osób, która poprawnie odpowie [redacted] poniższe pytania. Mamy nadzieję, [redacted] więcej tak ekspansywnych firm. Nasze łamy stoją dla Was zawsze otworem i każdy kto chce coś u [redacted] napisać, coś zaprezentować, czy sponsorować jakiś Quiz, czy konkurs jest mile widziany. Wraz z naszymi czytelnikami czekamy [redacted] listy i propozycje!

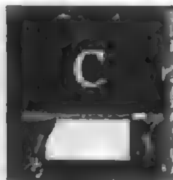
A teraz już [redacted] [redacted] pytania. Na odpowiedzi będziemy czekali przez miesiąc. Wśród tych, którzy nadeślą prawidłowe odpowiedzi rozlosujemy paczkę dysków ufundowaną przez Twin Spark Soft (Postaramy się nagrody nie dzielić, he, he).

1. Podaj chociaż trzy główne różnice między A500 [redacted] A500+.
2. Z jaką podstawową konfiguracją pamięci sprzedawane były Amigi [redacted] przestrzeni dziejów?
3. Czym FIZYCZNIE różni się dyskietka jednostronna od dwustronnej?
4. Jaka firma "podpisuje się" pod najbardziej znanymi grami symulacyjnymi (np. PIRATES, RAILROAD TYCOON, KNIGHTS OF THE SKY)
5. Podaj chociaż trzy tytuły gier firmy CINEMAWARE.

A może Wy macie pomysł [redacted] Quiz, czy konkurs? Może chcielibyście coś takiego zorganizować [redacted] Waszym osiedlu i przysłać nam relację z tego wydarzenia, podając np. komisyjnie stwierdzone najlepsze czasy w Lotusie itp?

Może Ty [redacted] pomysł na Amigową Listę przebojów? Jak ją zrealizować, by była atrakcyjna i by wszyscy byli [redacted] niej zadowoleni?

Redakcja



Kurs C cz.3

W dzisiejszym odcinku Kursu C zajmiemy się w dalszym ciągu grafiką - w tym dużo miejsca zajmą wszystkim dobrze znane literki. (No, bo którz nie styka się z nimi na codzień?).

T.K.

VII: NASZE PIERWSZE LITERKI

No tak, rysować już potrafimy. Ale co z podpisami pod rysunkiem? Ci z Was, którzy nie ograniczyli się do przepisywania programów z poprzednich odcinków, ale próbowali eksperymentowania zauważyli [redacted] pewnością dziwną rzecz: tekst wypisywany [redacted] pomocą funkcji Text umieszczany jest [redacted] rastporcie w dość dziwnym miejscu - 6 pikseli [redacted] punktem wyznaczonym przez funkcję **Move**. Dlaczego tak jest?

Spójrzmy jeszcze [redacted] strukturę [redacted] Widzimy tam trzy komponenty: **TxWidth**, **TxHeight** i **TxBaseLine** odnoszące się do rozmiarów pisanych znaków. **TxWidth** i **TxHeight** to po prostu szerokość i wysokość pojedynczego [redacted] (możemy łatwo sprawdzić, że w przypadku topaz/8 będą [redacted] ósemki).

Cóż to jest natomiast **TxBaseLine**?

Otóż każdy font (czyli zestaw znaków) jest wyświetlany nie od lewego górnego punktu litery, lecz od linii bazowej w górę i w dół. Umożliwia to łatwe wyświetlanie tekstów za pomocą różnych fontów. I tak np. *topaz/8* [redacted] wartość linii bazowej równą właśnie 6 - co oznacza, że każda litera może wystawać w górę o 6 pikseli.

Jeśli więc chcemy np. podpisać rysunek w ten sposób, aby lewy górny róg tekstu wypadał na pozycji (x,y), należy napisać instrukcję typu:

```
Move(RP,x,y+RP-> TxBaseLine);
```

Dodatkowo (jak może również niektórzy zauważyli) występuje problem pisania w [redacted] nie posiadających znacznika: GIMMEZEROZERO - po prostu jeśli się nie uważa, można wyleźć [redacted] ramkę. Co prawda można sobie ustalić, że nie będziemy wyłazić poza punkt (4,12) [redacted] przykład, ale co zrobić w przypadku, jeśli akurat piszemy nową, poprawioną wersję DeluxePaint-VI i okno [redacted] ramkę a [redacted] jej nie ma? Użycie okna GIMMEZEROZERO, choć [redacted] pozoru wygodne, [redacted] też swoje wady: po prostu spowalnia wszystkie operacje graficzne.

```
LISTING # 1:
  #include <intuition/intuition.h>
  #include <functions.h>

  struct Screen *Screen;
  struct Window *Window;
  struct IntuitionBase *IntuitionBase;
  struct GfxBase *GfxBase;

  struct TextAttr AFont={
    (UBYTE *)"topaz.font",
    8,0,FPF_ROMFONT
    };
  struct NewScreen ekran={
    0,0,320,200,2,0,1,0,WBENCHSCREEN,
    &AFont,(STRPTR)"Moj ekran",0,0};

  struct NewWindow okno={
    0,0,320,200,0,1,0,
    ACTIVATE,
    0,0,(STRPTR)"Moje okienko",0,0,
    100,100,320,200,CUSTOMSCREEN};

  int Lewy,Gorny,Wysokosc;
  char linia[40];

  main()
  {
    IntuitionBase=(struct IntuitionBase *)
      OldOpenLibrary("intuition.library");
    if (!IntuitionBase) exit(100);
    GfxBase=(struct GfxBase *)
      OldOpenLibrary("graphics.library");
    if (!GfxBase) exit(100);
    if(!(Screen=OpenScreen(&ekran))) koncz();
    okno.Screen=Screen;
    if(!(Window=OpenWindow(&okno))) koncz();

    Lewy=Window-> BorderLeft;
    Gorny=Window-> BorderTop+Window-> RPort
-> TxBaseline;
    Wysokosc=Window-> RPort-> TxHeight;

    sprintf(linia,"Wysokosc=%d, Szerokosc=%d",
      Window-> RPort-> TxHeight,
      Window-> RPort-> TxWidth);
    Move(Window-> RPort,Lewy,Gorny);
    Text(Window-> RPort,linia,strlen(linia));

    sprintf(linia,"Linia bazowa=%d",
      Window-> RPort-> TxBaseline);
    Move(Window-> RPort,Lewy,Gorny +Wysokosc);
    Text(Window-> RPort,linia,strlen(linia));

    sprintf(linia,"Szczyt tekstu [redacted] x %d",
      Lewy,Gorny);
    Move(Window-> RPort,Lewy,Gorny+2*Wysokosc);
    Text(Window-> RPort,linia,strlen(linia));

    czekaj();
    koncz();
  }

  czekaj()
  {
    while (Window-> MouseX> 10 [redacted]
Window-> MouseY> 10) Delay(1);
  }

  koncz()
  {
    if (Window) CloseWindow(Window);
    if (Screen) CloseScreen(Screen);
    CloseLibrary(IntuitionBase);
    CloseLibrary(GfxBase);
    exit(0);
  }
```

Zajrzyjmy do struktury Window. Mamy tam m.in. komponenty [redacted] nazwie **BorderLeft** i **BorderTop**, czyli po prostu aktualne położenie lewego górnego rogu wolnego miejsca [redacted] oknie. A więc nasza instrukcja może w tej chwili wyglądać następująco:

```
Move(Window-> RPort,
x+Window-> BorderLeft,
y+Window-> BorderTop+Window
-> RPort-> Baseline);
```

Mamy teraz stuprocentową pewność, [redacted] tekst będzie umieszczony dokładnie tam, gdzie chcemy względem lewego górnego rogu okna.

Jasne?

A więc próbujemy:

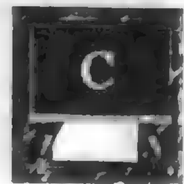
(Listing # 1)

VIII: ZABAWA W STYLU BOLD, CZYLI WYGINAMY FONTA

Jak powszechnie wiadomo, każda (no, prawie) litera może być pisana [redacted] kilka różnych sposobów. Może być podkreślona, *pochylona*, **pogrubiona** albo też styl pisania może być tych sposobów kombinacją.

Taka zmiana stylu reali[redacted] jest w Amidze programowo - każdy znak (a ściślej jego obraz graficzny) przed "wyrzuceniem" [redacted] na rastport może być poddany pewnej obróbce. Dzieje się to automatycznie przy wykonywaniu funkcji **Text**, należy jednak komputer poinformować, jakiego stylu ma użyć. Robimy to przy pomocy funkcji **SetSoftStyle**.

Funkcja [redacted] trzy parametry. Pierwszym z nich jest oczywiście rastport, drugi to kombinacja stylów, których chcemy użyć, natomiast trzeci to maska zezwolenia. Jeśli chodzi o style, mamy ich do wyboru trzy, każdy sterowany oddzielnym bitem:



FSF_ITALIC
- kursywa
FSF_UNDERLINED
- podkreślenie
FSF_BOLD
- pogrubienie.

Maska zezwolenia służy do poinformowania, które [illegible] stylów mogą być użyte. Istnieją dwie możliwości wykorzystania maski. Pierwsza z nich to próba zmiany np. [illegible] kursywę fonta, który z założenia jest kursywą. Taka zmiana nie powinna być wykonana. Dla sprawdzenia możemy zapytać o dostępne style [illegible] pomocą funkcji **AskSoftStyle**:
(Listing # 2)

```
LISTING # 2
#include <intuition/intuition.h>
#include <functions.h>

struct Screen *Screen;
struct Window *Window;

struct IntuitionBase *IntuitionBase;
struct GfxBase *GfxBase;

struct TextAttr AFont={
   (UBYTE *)"topaz.font",
   8,0,FPF_ROMFONT
   };
struct NewScreen ekran={
   0,0,320,200,2,0,1,0,WBENCHSCREEN,
   &AFont,(STRPTR)"Moj ekran",0,0};

struct NewWindow okno={
   0,0,320,200,0,1,0,
   ACTIVATE|BORDERLESS,
   0,0,0,0,0,
   100,100,320,200,CUSTOMSCREEN};

main()
{

   IntuitionBase=(struct IntuitionBase *)
      OldOpenLibrary("intuition.library");
   if (!IntuitionBase) exit(100);
   GfxBase=(struct GfxBase *)
      OldOpenLibrary("graphics.library");
   if (!GfxBase) exit(100);
   if(!(Screen=OpenScreen(&ekran))) koncz();
   okno.Screen=Screen;
   if(!(Window=OpenWindow(&okno))) koncz();

/* ustawianie stylow */

   SetSoftStyle(Window->RPort,FSF_UNDERLINED,255);
   PrintAt(40,40,"Tekst podkreslony");
   SetSoftStyle(Window->RPort,FSF_BOLD,255);
   PrintAt(40,50,"Tekst pogrubiony");
   SetSoftStyle(Window->RPort,FSF_BOLD|FSF_ITALIC,255);
   PrintAt(40,60,"Pogrubiona kursywa");

/* maskowanie stylow */

   SetSoftStyle(Window->RPort,FSF_UNDERLINED,255);
   PrintAt(40,80,"Podkreslenie");
   SetSoftStyle(Window->RPort,FSF_BOLD,FSF_BOLD);
   PrintAt(40,90,"Pogrubienie z podkresleniem");
   SetSoftStyle(Window->RPort,0,FSF_BOLD);
   PrintAt(40,100,"I znow tylko podkreslenie");

   czekaj();
   koncz();
}

czekaj()
{
   while (Window->MouseX>10 &&
Window->MouseY>10) Delay(1);
}

koncz()
{
   if (Window) CloseWindow(Window);
   if (Screen) CloseScreen(Screen);
   CloseLibrary(IntuitionBase);
   CloseLibrary(GfxBase);
   exit(0);
}

PrintAt(x,y,t) int x,y;char *t;
{
   Move(Window->RPort,x,y);
   Text(Window->RPort,t,strlen(t));
}
```

Myślę, że opanowanie zmian stylu nie sprawi większej trudności. Dodać tu mogę jeszcze, że **SetSoftStyle** zwraca jako wynik aktualny styl pisania.

IX: PISMO ŚWIADCZY O CZŁOWIEKU

Fajnie. Możemy sobie popisać. Nawet jeśli mieliśmy trochę inwencji, mogliśmy sobie wybrać pomiędzy *topaz/8* [illegible] *topaz/9*. Tyle, że dla naszych potrzeb to trochę [illegible] mało - przez cały okres "życia" okna jesteśmy związani z fontem, którego zażądaliśmy otwierając okno. Musimy mieć możliwość zmiany fonta w trakcie pracy - dają nam ją funkcje **OpenFont**, **SetFont** i **CloseFont**.

Parametrem funkcji OpenFont jest wskaźnik do odpowiedniej struktury **TextAttr**, zawierającej informacje o foncie, który będziemy otwierać. Struktura [illegible] wygląda następująco:

```
struct TextAttr {
  STRPTR  ta_Name;
  UWORD   ta_YSize;
  UBYTE   ta_Style;
  UBYTE   ta_Flags;
  }
```

ta_Name to [illegible] fonta (koniecznie z przyrostkiem ".font"!), **ta_YSize** to wysokość, **ta_Style** to programowy styl pisania natomiast **[illegible]_Flags** zawiera różne dodatkowe informacje o foncie.

Nas interesuje na razie tylko jedna: jest nią FPF_ROMFONT - znak, że font znajduje się w ROMie Amigi.

Za pomocą **OpenFont** możemy otworzyć *topaz/8* lub *topaz/9* - jako, że tylko one znajdują się w ROMie. W tym celu musimy zainicjalizować strukturę i wywołać funkcję **OpenFont**, np. tak:

```
struct TextAttr TA={(STRPTR)
topaz.font",8,0,FPF_ROMFONT};
 struct TextFont *TF;
 ...

 TF=OpenFont(&TA);
```

Struktura TextFont, do której wskaźnik zwraca **OpenFont**, zawiera wszystko, co jest potrzebne Amidze do pisania, dla nas natomiast jest raczej mało interesująca.

Jeśli font jest już otwarty, możemy przydzielić do odpowiedniego rastportu. Jeśli jednak będziemy często zmieniać fonty, warto zapamiętać ten, który był przydzielony do rastportu przed zmianami i przed każdym zamknięciem nowego fonta przydzielić stary z powrotem do rastportu. Struktura **RastPort** zawiera komponent:

```
struct TextFont *Font
```

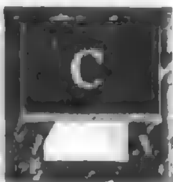
którego wartością jest font przydzielony do rastportu.

Samo przydzielenie jest proste - wywołujemy w tym celu funkcję **SetFont** (RP,Font) i to wszystko.

Jeśli font nie jest nam już potrzebny, możemy zamknąć go funkcją CloseFont. Dodatkowo, jeśli chcemy uzyskać parametry fonta przydzielonego do rastportu, możemy użyć funkcji **AskFont**, której parametrami [illegible] rastport i wskaźnik do pustej struktury **TextAttr**. W wyniku działania funkcji struktura ta zostanie zapełniona odpowiednimi danymi.

Tak więc zobaczmy to w praktyce:
(Listing # 3).

Piszesz programy? Pisz do nas!



```
LISTING # 3
#include <intuition/intuition.h>
#include <functions.h>

struct Screen *Screen;
struct Window *Window;

struct IntuitionBa[redacted] *IntuitionBase;
struct GfxBase *GfxBase;

struct TextAttr AFont={
  (UBYTE *)"topaz.font",
  8,0,FPF_ROMFONT
  };

struct TextAttr BFont={
  (UBYTE *)"topaz.font",
  9,0,FPF_ROMFONT
  };

struct NewScreen ekran={
  0,0,320,200,2,0,1,0,WBENCHSCREEN,
  &AFont,(STRPTR)"Moj ekran",0,0};

struct NewWindow okno={
  0,0,320,200,0,1,0,
  ACTIVATE|BORDERLESS,
  0,0,0,0,0,
  100,100,320,200,CUSTOMSCREEN};

struct TextFont *TF8,*TF9,*OldFont;
char linia[40];

main()
{

  IntuitionBase=(struct IntuitionBase *)
    OldOpenLibrary("intuition.library");
  if (!IntuitionBase) exit(100);
  GfxBase=(struct GfxBase *)
    OldOpenLibrary("graphics.library");
  if (!GfxBase) exit(100);
  if(!(Screen=OpenScreen(&ekran))) koncz();
  okno.Screen=Screen;
  if(!(Window=OpenWindow(&okno))) koncz();

  OldFont=Window->RPort->Font;

  TF8=OpenFont(&AFont);
  TF9=OpenFont(&BFont);

  SetFont(Window->RPort,TF8);
  GetFont();
  PrintAt(10,20,linia);

  SetFont(Window->RPort,TF9);
  GetFont();
  PrintAt(10,30,linia);

  czekaj();
  SetFont(Window->RPort,OldFont);
  CloseFont(TF8);
  CloseFont(TF9);
  koncz();
}

GetFont()
{
  struct TextAttr TA;
  AskFont(Window->RPort,&TA);
  sprintf(linia,"%s %d",TA.ta_Name,TA.ta_YSize);
}

czekaj()
{
  while (Window->MouseX>10 [redacted] Window->MouseY>10)
Delay(1);
}

koncz()
{
  if (Window) CloseWindow(Window);
  if (Screen) CloseScreen(Screen);
  CloseLibrary(IntuitionBase);
  CloseLibrary(GfxBase);
  exit(0);
}

PrintAt(x,y,t) int x,y;char *t;
{
  Move(Window->RPort,x,y);
  Text(Window->RPort,t,strlen(t));
}
```

No dobrze, ale w ten sposób możemy wybierać tylko między *topaz/8* [redacted] *topaz/9*. Do naszych celów to stanowczo [redacted] mało! Aby użyć fontów znajdujących się [redacted] dysku, musimy otworzyć na-tępną bibliotekę - ***diskfont.library***.

W przeciwieństwie do intuition.library i graphics.library, biblioteka [redacted] musi znajdować się [redacted] dysku w systemowym katalogu LIBS:.

Funkcja **OpenDiskFont** udostępnia [redacted] fonty znajdujące się w katalogu FONTS:. Użycie jest identyczne jak **OpenFont**, tak więc od [redacted] przykład:

(Listing # 3)

I tu uwaga dla posiadaczy bardzo starych Amig z wersją *Kickstartu 1.1*: w systemie tym znajduje się błąd polegający [redacted] tym, że zamknięty font może zostać skasowany z pamięci, ale system [redacted] tym nie wie! Tak, że w tych komputerach nie należy używać funkcji [illegible]

Przy fontach ładowanych [redacted] dysku pojawia się dodatkowa trudność: o ile znajdujący się w ROMie *topaz* [redacted] stałą szerokość znaku, dostępną np. przez RP->TxWidth, [redacted] tyle np. *sapphire* jest tzw. fontem proporcjonalnym, w którym litery mają różną szerokość (np. "i" jest dużo węższe od "w"). W związku z tym możemy mieć problemy ze sprawdzeniem, czy dany napis zmieści się nam w szerokości rastportu, jako że **TxWidth** zawiera przybliżoną przeciętną szerokość znaku. Do obliczenia długości tekstu w pikselach służy funkcja:

```
Dlugosc =
TextLength(RP,tekst,ilosc_znakow);
```

Proste i nie wymaga omawiania.

X:COŚ DLA ARTYSTÓW...

Wróćmy do rysowania. Niekiedy zachodzi konieczność narysowania linii nie ciągłej, lecz przerywanej (np. rysunki techniczne). Amiga [redacted] taką możliwość realizowaną w dodatku sprzętowo przez blittera. Otóż rastport zawiera pole **LinePtrn**, które służy do ustawiania wzorca, według którego rysowana będzie każda linia. W pliku "graphics/gfxmacros.h" znajduje się makrodefinicja SetDrPt, służąca do ustalania owego wzorca. Sam wzorzec natomiast jest szesnastobitową wartością, gdzie każdy bit odpowiada rysowanemu pikselowi. W miejscach, gdzie odpowiednie bity [redacted] równe 0, linia nie będzie rysowana. Jeśli np. chcemy [redacted] oknie Window narysować linię kropkowo-przerywaną, możemy to zrobić wywołując:

```
SetDrPt(Window->RPort,0x18ff);
```

gdzie liczba 0x18ff odpowiada dwójkowemu 0001100011111111.

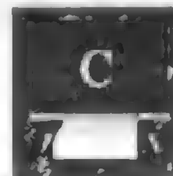
Podobnie dzieje się z malowaniem powierzchni. Tak jak dla linii, tak i dla funkcji malujących (a więc **RectFill**, **Flood**, **AreaEnd**) możemy ustanowić wzorzec, według którego powierzchnia będzie zamalowywana.

Wzorzec taki [redacted] szerokość 16 bitów, natomiast jego wysokość może być dowolną potęgą dwójki. Przydzielić wzorzec do rastportu możemy przez np:

```
UWORD Pattern[]={
  0x8888,0x4444,
  0x2222,0x1111
  };
  ...

SetAfPt(Window
->RPort,Pattern,2);
```

I tu uwaga: ostatni parametr **SetAfPt** jest wykładnikiem potęgi dwójki tworzącej wysokość wzorca, [redacted] nie wysokością!




```
LISTING # 4
#include <intuition/intuition.h>
#include <graphics/gfxmacros.h>
#include <functions.h>

struct Screen *Screen;
struct Window *Window;

struct IntuitionBase *IntuitionBase;
struct GfxBase *GfxBase;

struct TextAttr AFont={
  (UBYTE *)"topaz.font",
  8,0,FPF_ROMFONT
  };
struct NewScreen ekran={
  0,0,320,200,1,0,1,0,WBENCHSCREEN,
  &AFont,(STR■TR)"Moj ekran",0,0};

struct NewWindow okno={
  0,0,320,200,0,1,0,
  ACTIVATE|BORDERLESS,
  0,0,0,0,0,
  100,100,320,200,CUSTOMSCREEN};

WORD walek1[28]={
0,28,8,36,32,36,32,86,48,86,48,32,68,32,
68,24,80,24,80,32,120,32,130,42,130,82,150,82};
WORD walek2[28];
UWORD Kreski[]={
  0x8888,0x4444,0x2222,0x1111};
main()
{
  int i;
  IntuitionBase=(struct IntuitionBase *)
    OldOpenLibrary("intuition.library");
  if (!IntuitionBase) exit(100);
  GfxBase=(struct GfxBase *)
    OldOpenLibrary("graphics.library");
  if (!GfxBase) exit(100);
  if(!(Screen=OpenScreen(&ekran))) konoz();
  okno.Screen=Screen;
  if(!(Window=OpenWindow(&okno))) konoz();

  AddTmpRas(Window->RPort,320,200);

  for (i=0;i<28;i+=2) {
    walek1[i]+=30;
    walek2[i]=walek1[i];
    walek2[i+1]=100-walek1[i+1];
    walek1[i+1]+=100;
    }

  SetAPen(Window->RPort,1);
  SetOPen(Window->RPort,1);
  S■tAfPt(Window->RPort,Kreski,2);
  Ar■aMove(Window->RPort,30,100-16);
  for (i=0;i<28;i+=2) AreaDraw(Window
->RPort,walek2[i],walek2[i+1]);
  AreaDraw(Window->RPort,walek2[26],100-16);
  Ar■aEnd(Window->RPort);

  Move(Window->RPort,30,100-16);
  PolyDraw(Window->RPort,14,walek1);
  for (i=0;i<28;i+=2) {
    Move(Window->RPort,walek1[i],walek1[i+1]);
    Draw(Window->RPort,walek1[i],100);
    }
  Move(Window->RPort,walek1[26],walek1[27]);
  Draw(Window->RPort,walek1[26],100-16);
  SetDrPt(Window->RPort,0x18ff);
  Move(Window->RPort,15,100);
  Draw(Window->RPort,walek2[26]+15,100);
  czekaj();
  FreeTmpRas(Window->RPort);
  konoz();
}

czekaj()
{
  while (Window->MouseX>10 && Window-
>MouseY>10) Delay(1);
}

konoz()
{
  if (Window) CloseWindow(Window);
  if (Screen) CloseScreen(Screen);
  CloseLibrary(IntuitionBase);
  CloseLibrary(GfxBase);
  exit(0);
}
```

Dlatego dla 4-liniowego wzorca użyłem 2 (bo $2^2=4$...).

Przykład jest prosty. Nie zapomnijmy tylko ■ funkcjach AddTmpRas i FreeTmpRas (patrz poprzedni odcinek).

(Listing # 4)

<u>XI: ...I PROGRAMISTóW</u>

Pamiętacie pewnie funkcję ClipBlit służącą do przenoszenia fragmentów rysunku. Wróćmy jeszcze ■ do ostatniego parametru tej funkcji. Jest to po prostu kod pracy (tzw. minterm) blittera składający się z czterech bitów.

Blitter przenosi zawartość jednego planu do drugiego punkt po punkcie, ale wynikowy punkt jest obliczany ■ pomocą określonej funkcji z punktów na planie źródłowym (plan B) oraz wynikowym (plan C). Cztery bity funkcji oznaczają:

0x10 C=(~B)&(~C)
0x20 - C=(~B)&C
0x40 - C=B&(~C)
0x80 - C=B&C

Połączenia tych wartości dają sumę logiczną funkcji, i tak np. dla podanej uprzednio wartości 0xc0 (czyli 0x80 | 0x40) mamy:

Przykładowego programu nie podaję - ■ tym etapie każdy powinien napisać go sam!

I to ■ dzisiaj tyle.
Życzę przyjemnego rysowania!

Bohdan R.Rau

A oto kilka innych pożytecznych przykładów:

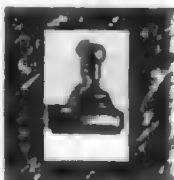
0x30:
C=((~B)&C)|((~B)&(~C))=(~B)&(C|(~C))=~B
Wynik jest zanegowaną kopią źródła.

0x50:
C=(B&(~C))|((~B)&(~C))=(~C)&(B|(~B))=~C
Wynik jest swoją własną negacją

0x60: C=(B&(~C))|((~B)&C)
Wynik jest różnicą symetryczną źródła i wyniku

Przypominamy o naszym konkursie na najprzydatniejszy program. Każdy nadesłany do nas program, będzie oceniony przez naszych 'fachowców' i jeśli uznamy, że jest przydatny dla szerszego grona czytelników, zostanie on opublikowany na łamach AMIGOWCA, zaś jego autor otrzyma autorskie honorarium. Wysokość honorariów, będzie bardzo zróżnicowana, w zależności od 'kalibru' programu. Oczywiście jeżeli wysokość ustalona przez Nas nie będzie Wam odpowiadała, nie opublikujemy wbrew Waszej zgodzie kodu źródłowego programu, ani sam program nie zostanie zamieszczony na naszym dysku PD. Czekamy na Waszą twórczość.

Redakcja



Deuteros

- The Next Millenium.

Wszyscy pamiętacie zapewne świetną grę strategiczną pt. *Millenium 2.2* firmy *Activision*. Jeśli udało się ją Wam ukończyć, to wiecie, że życie [redacted] Ziemi po uderzeniu wielkiego asteroidu odrodziło się. Minęło kilka stuleci i Ludzie znowu zainteresowali się Księżycem i planetami Układu Słonecznego. Przypomniano sobie o opuszczonej kolonii [redacted] Księżycu i lotach kosmicznych. Rozpoczęła się Operacja Deuteros. Ale życie rozwijało się nie tylko [redacted] Ziemi. W dawnej kolonii na planecie Saturn nastąpiła mutacja i rozwinęła się [redacted] Metanoidów, która także podjęła kosmiczną ekspansję. Konflikt obu ras wydaje się nieunikniony. W tym momencie wkrac[redacted] Ty, Drogi Czytelniku i przejmujesz sterowanie i kontrolę nad centralnym komputerem w Earth City. Twoim zadaniem jest skolonizowanie Układu Słonecznego (a później innych systemów) pokonanie wroga. Centralny Komputer został tak, skonstruowany, żeby umożliwić kierowanie wszystkimi koloniami, statkami kosmicznymi i komunikacją. Poprzez klikanie [redacted] poszczególne ikony otrzymujemy informacje o stanie magazynów, na planetach, stacjach orbitalnych i o załogach.

Przyspieszacz [redacted].

Na dole ekranu znajduje się wskaźnik upływu czasu, który podaje nam aktualną datę i godzinę. Ponieważ produkcja i wydobycie surowców zajmuje trochę czasu, po ustawieniu parametrów możemy przyspieszyć jego upływ, dopóki odpowiednia ilość surowca nie zostanie wydobyta, bądź też produkcja nie zostanie ukończona. Bieg czasu wraca do normy po ponownym wciśnięciu lewego przycisku myszy.

Operacje dyskowe.

Poprzez kliknięcie tej ikony wchodzimy do menu operacji dyskowych. Mamy możliwość sformatowania dysku, zapisania stanu gry lub też wczytania naszych ostatnich osiągnięć.

Panel informacyjny.

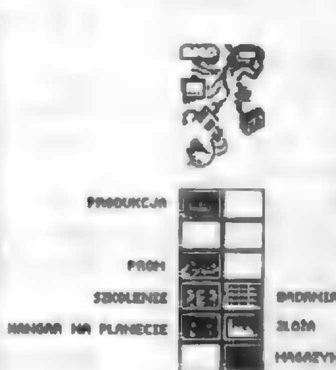
Po uruchomieniu tej opcji ukazują się przed nami najważniejsze informacje. Przy każdej znajduje się data i [redacted] pojawienia. W początkowej fazie gry regularne oglądanie informacji nie jest potrzebne, jednak gdy rozbudujemy stacje w wielu systemach planetarnych kontrola bieżących wydarzeń staje się niezbędna.

Earth City.

Kliknięcie tej ikony powoduje natychmiastowy powrót do Earth City.

Sterowanie główne.

Ikona pojawia się gdy zbudujemy pierwszą fabrykę w przestrzeni. Jej kliknięcie powoduje ukazanie się listy wszystkich kolonii i statków międzyplanetarnych. Kliknięcie jakiejkolwiek ikony powoduje natychmiastowy dostęp do danego statku lub stacji orbitalnej. Do stacji znajdujących się [redacted] planetach mamy dostęp tylko przez stacje orbitalne.





Analiza złóż dostępnych pierwiastków.

Ta' ikona [redacted] znaczenie wyłącznie informacyjne. Po zaktywizowaniu ukazuje się mapa Układu Słonecznego, identyczna z [redacted] systemu nawigacyjnego statku międzyplanetarnego. Poniżej mapy znajduje się spis pierwiastków dostępnych na aktualnie wskazanej planecie względnie jej księżycach.

Centralny bank danych.

Dostarcza informacji o surowcach i produktach we wszystkich stacjach kosmicznych. Inna lista w nim dostępna zawiera spis teamów produkcyjnych, naukowców i pilotów. Dostępne [redacted] także dalsze opcje, które służą do przełączenia pomieszczenia magazynowego np:
-Magazyn surowców;
-Magazyn produktów;
-Spis zespołów roboczych.

Produkcja.

Kliknięcie ikony powoduje ukazanie się ekranu sterowania produkcją. Zawiera on panel wyboru przedmiotów do produkcji, wskaźnik zaawansowania produkcji, [redacted] status zespołu produkcyjnego. Aby móc coś wyprodukować, musimy najpierw ściągnąć zespół produkcyjny z hangaru względnie z centrum szkoleniowego z Ziemi. Produkcja większości przedmiotów wymaga od [redacted]fów zespołów produkcyjnych pewnego doświadczenia. Im większe jest doświadczenie i tytuł szefa, tym więcej i szybciej można produkować. Ilość osób (max. 200) w zespole także ma wpływ [redacted] szybkość produkcji. Produkcja każdego przedmiotu wymaga odpowiednich surowców. Niedostateczna ich ilość jest sygnalizowana odpowiednim komunikatem. Jeśli żadnych surowców nie brakuje, rozpoczyna się produkcja. Gotowe przedmioty [redacted] natychmiast przeno[redacted] do magazynu. Musimy przy tym pamiętać, że niektóre przedmioty nie mogą być transportowane (np. Auto Operations Computer) i produkowanie ich w większych ilościach w jednym miejscu jest tylko stratą [redacted] i surowców.

Magazyny.

Earth City, fabryki [redacted] stacjach orbitalnych i stacje naziemne posiadają własne magazyny surowców i gotowych wyrobów. Między tymi magazanami możemy przełączać, klikając służącą do tego celu ikonę. Magazyn surowców ukazuje listę



wszystkich możliwych surowców. Po lewej stronie pokazane są ilości surowców, które aktualnie leżą w magazynach. Niebieskie strzałki sygnalizują, jakie surowce znajdują



się na planecie (księżycu). po prawej stronie znajduje się panel wyboru przedmiotów. Kliknięcie na którykolwiek przedmiot powoduje ukazanie się informacji, ile sztuk można wyprodukować z posiadanej ilości materiałów, względnie ewentualne braki surowców. Magazyn wyrobów gotowych zawiera listę wszystkich przedmiotów razem z aktualnym stanem magazynu. Magazyn może pomieścić max. 15 sztuk każdego produktu. Wybranie którejkolwiek nazwy powoduje ukazanie się krótkiego opisu funkcji każdego urządzenia.

Złoża surowców.

Dostęp do złóż surowców mamy w Earth City i wszystkich stacjach naziemnych. Kliknięcie ikony powoduje przejście do ekranu, z którego możemy odczytać przewidywaną (przez centralny komputer) ilość surowców, jaką można wydobyć na planecie (księżycu).

Centrum szkoleniowe.

Dostępne jest tylko w Earth City. Możemy w nim szkolić nowych pracowników, pilotów i naukowców. Po przydzieleniu odpowiedniej ilości osób do zespołów, przyspieszamy upływ czasu i szkolenie rozpoczyna się. Gdy zostanie ukończone, zespoły zostają automatycznie przeniesione do hangaru, gdzie oczekują na przydział. Szef każdego zespołu ma początkowo najniższą możliwą rangę. Może ją podwyższyć przez praktykę. Zespół badawczy może liczyć max. 250 osób, produkcyjny 200, a pilotów 41 osób.

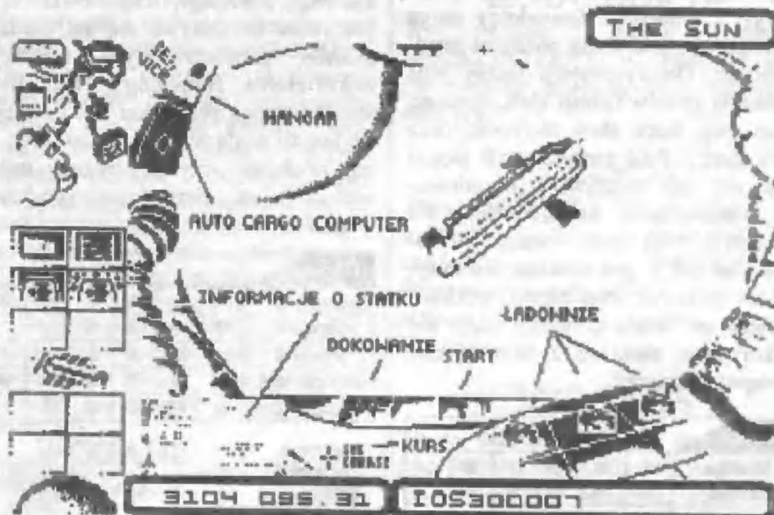
Centrum naukowe.

Mamy do dyspozycji tylko jedno i znajduje się ono w Earth City. Jego sprawność i efektywność zależy od liczebności i stopnia szefa naukowego. Nauko-zieleń oznacza, że projekt jest gotowy do produkcji.

Hangar.

Istnieją dwa rodzaje hangarów. Jeden jest przeznaczony dla promów prowadzących komunikację na trasie planeta-stacja orbitalna i drugi dla statków międzyplanetarnych. Ten drugi mamy do dyspozycji tylko na stacjach orbitalnych, ponieważ statki międzyplanetarne nie mogą lądować na planetach. Pusty hangar może być użyty do budowy nowego statku. W tym celu należy kliknąć ikonę *New Ship*, co spowoduje wstawienie przedziału załogowego do hangaru. Teraz należy tylko zamontować silnik i ładownie, przydzielić załogę (poprzez kliknięcie na wybrany zespół) i można już ruszać na podbój przestrzeni. Tu krótka uwaga, kliknięcie zespołu produkcyjnego powoduje jego przeniesienie do fabryki, a nie do kabiny pilotów.

Istnieją trzy rodzaje ładowni. Jedna służy do przewozu gotowych wyrobów, druga do surowców, trzecia zaś do przewozu ludzi. Aby zamontować daną ładownię, trzeba kliknąć najpierw odpowiedni przedział statku na piktogramie u góry ekranu, a następnie odpowiednią ikonę ładowni u dołu z prawej strony.



Statek kosmiczny.

Kiedy znajdujemy się w hangarze, aby przejąć kontrolę nad statkiem, klikamy na przedział załogowy. Powoduje to przejście do kokpitu statku. Klikamy teraz ikonę ustawienia kursu, wybieramy punkt docelowy, po czym wystrzeliwujemy statek ze stacji orbitalnej, odpalamy silnik i naprzód! Kiedy statek znajduje się w przestrzeni, bądź też na orbicie, wystarczy kliknąć jego ikonę na ekranie sterowania głównego, aby znaleźć się w kokpicie. Pamiętajcie o uprzednim wyłączeniu silnika, jeśli chcecie zmienić kurs statku podczas lotu. Oszczędzi to wiele czasu i paliwa. Powyższe uwagi dotyczą statku międzyplanetarnego. Obsługa promu orbitalnego jest znacznie prostsza, gdyż odpada ustawianie kursu i odpalanie silników. Cała procedura ogranicza się do startu (lądowania) z powierzchni planety i wystrzelenia ze stacji (dokowania). Uwaga! Na każdej planecie (księżycu) może być czynny tylko jeden prom.

Sądzę, że powyższy krótki opis stanie się dużą pomocą dla naszych Czytelników skazanych zwykle na wielogodzinne "rozgryzanie" co lepszych i bardziej skomplikowanych gier. Tu jeszcze jedna uwaga. Na giełdach krąży kilka wersji Deuteros`a. Wersja jednodyskowa to tylko demo, które umożliwia grę do pewnego momentu. Oryginał ma dwa dyski, natomiast ja znalazłem na giełdzie wersję trzydyskową, z której (prawdopodobnie!!!) tylko dwa dyski są do czegoś potrzebne. Dysk trzeci zawiera tylko w wielu miejs-cach komunikat o zabezpieczeniu jakimś antywirusem i nic poza tym.

Radzę więc wszystkim zaopatrzyć się właśnie w tę wersję (a jeszcze lepiej w oryginał).

Życzę przyjemnej zabawy

Tomasz Kulbacki.





TIP OFF

ANCO

Sławna z cyklu gier pt. KICK OFF firma ANCO wypuściła na rynek gier komputerowych swój najnowszy produkt: grę TIP OFF. Jest to nowa wersja koszykówki. Gry sportowe cieszą się ogromnym powodzeniem wśród graczy i dlatego istnieje ich cała masa. Kilka lat temu firma ANCO wprowadziła na rynek grę KICK OFF. Była to piłka nożna i niemal od razu gra ta stała się przebojem. Programiści perfekcyjnie dostosowali zasady i dynamikę gry do jej komputerowego odpowiednika i jest obecnie swoistym wzorcem dla tych wszystkich programistów, którzy próbują stworzyć symulację gry zespołowej. TIP OFF jest doskonałym tego przykładem. Otwiera ona przed nami wiele możliwości dotyczących zarówno samej gry, jak i wybór składów, poziomów trudności itp. Gracz może sam ustalać cechy poszczególnych zawodników swojej drużyny (wiek, wysokość, energia, siła, wytrzymałość) oraz ich umiejętności (drybling, wyskok, "zbiórki", "przechwyty", celność). Zawodnicy mogą poruszać się z dwoma różnymi prędkościami. Do dyspozycji mamy pięć rodzajów rzutów (jump shot, running jump shot, hook shot, skyhook, oraz slam dunk). Pięć niezależnych stopni trudności daje możliwość stopniowego doskonalenia umiejętności. Na podstawie tych cech można powiedzieć, że jest to gra idealna. Niestety, tak nie jest. Jest ona bardzo trudna i dopiero po wielu próbach udaje się wykorzystać niektóre z teoretycznie dostępnych zagrań.

Ocena	bardzo dobra					
Grafika	*	*	*	*	*	
Dźwięk	*	*	*	*	*	
Pomysł	*	*	*	*	*	
Motywacja	*	*	*			
OGÓLNIE	❁	❁	❁	❁	❁	



FLIGHT OF THE INTRUDER

MIRRORSOFT

Przenosi nas w wir zmagań bojowych toczących się nad Północnym Wietnamem. Twoim zadaniem jest wspieranie naziemnych działań armii amerykańskiej oraz zwalczanie północno wietnamskich sił powietrznych. Tak przedstawia się fabuła nowego symulatora lotu firmy MIRRORSOFT. Jak każdy symulator lotu na Amigę tak i ten program szokuje doskonałą, niesamowicie szczegółową grafiką 3D. Ma jednak kilka dodatków, które stawiają go o wiele wyżej od innych symulatorów. Do wyboru mamy dwa najpopularniejsze w Wietnamie samoloty amerykańskie: F-4 Phantom (bardzo szybki, a przy tym potężny), oraz A-6 Intruder (wyposażony w sterowany komputerowo system uzbrojenia). Po przeciwnej stronie barykady mamy radzieckie samoloty MIG-17 i MIG-21. Nasze samoloty F-4 i F-6 znacznie różnią się od siebie. Inaczej wygląda ich kokpit, uzbrojenie i osiągi także są inne. Działania prowadzone są z pokładu lotniskowca. Do naszej dyspozycji są trzy rodzaje misji: MIGCAP, IRON HAND, STRIKE. Nowością jest jednoczesne działanie kilku amerykańskich samolotów między którymi występuje współdziałanie i przepływ informacji. Do zniszczenia mamy bardzo dużo różnorodnych obiektów np.: pociągi, ciężarówki itp. Wszystko to odbywa się na bardzo wiernie przedstawionych terenach wokół Hanoi, Haiphong i Than Hoa. Słabszą stroną programu jest dźwięk, ale jest to wadą wielu symulatorów, a tak właściwie czy koniecznie musi lecieć jakaś melodyjka podczas "lotu"?

Ocena	super!					
Grafika	*	*	*	*	*	*
Dźwięk	*	*	*	*		
Pomysł	*	*	*	*	*	*
Motywacja	*	*	*	*	*	*
OGÓLNIE	❁	❁	❁	❁	❁	ɛ



SHADOW DANCER

U.S.GOLD

Młody wojownik Ninja razem ze swoim wspaniałym, wiernym psem toczy bezlitosną walkę ze złem w centrum miasta, w którym policyjne patrole należą do rzadkości. Jego przeciwnikami są terroryści, których jednoczy wspólny cel chęć zawładnięcia miastem i bezlitosnego wykorzystania niewinnych obywateli. Taka jest fabuła kolejnej gry z popularnej serii "action games" czyli "bij, zabij". Nowością jest to, że nasz bohater nie walczy sam, ale dzielnie wspiera go pies. Gra ta nie wnosi nic ciekawego pod względem wątku. Również grafika, dźwięk oraz animacja nie zasługują na uwagę. Postacie są bardzo niewyraźne, a ruchy naszego wojownika wprost tragiczne. Całość przypomina nieco grę SHADOW WARRIOR i moim zdaniem jest bardzo słaba.

Ocena	dostateczna					
Grafika	*	*				
Dźwięk	*	*				
Pomysł	*					
Motywacja	*					
OGÓLNIE	❁	ɛ				



UTOPIA

Wszyscy Ci, którzy lubią grać w SIM CITY i MOONBASE powinni sięgnąć po najnowszą grę tego typu - pt. UTOPIA. Jej celem jest wybudowanie kolonii i utrzymywanie w niej jak najwyższego standardu życia. Standard ten pokazuje w jaki sposób prowadzona jest kolonia i jak poszczególne decyzje i działania wpływają na nastroje mieszkańców. Gra jest bardzo złożona i działania gracza muszą iść wieloma różnymi torami. Po pierwsze należy zbudować kolonię. Do dyspozycji mamy całą gamę budynków i obiektów o różnym przeznaczeniu. Są to np.: laboratoria wojenne, instalacje chemiczne, centra dowodzenia, szpitale, pasy startowe, kopalnie, radary, elektrownie, stadiony i wiele, wiele innych. Następnie trzeba zająć się ekonomią. Należy zwiększać bądź zmniejszać podatki oraz prowadzić rozległą działalność handlową. Kolejnym zadaniem jest kontrola rozwoju gospodarki. Trzeba dbać o rozwój różnych gałęzi przemysłu, o stan zatrudnienia oraz o wydajność posiadanych zakładów pracy. Innym kierunkiem naszej działalności jest infiltracja ras zamieszkujących daną planetę. Do zbierania informacji służą tajni agenci, których możemy wysyłać do "konkurencyj-



nych" kolonii. Kolejnym elementem gry jest walka z obcymi rasami. Do tego celu służą czołgi, statki kosmiczne, wieżyczki laserowe i wyrzutnie rakiet (istnieją również miny lądowe, ale trzeba je najpierw zbudować). Kolejnym ważnym czynnikiem, od którego zależy powodzenie naszej misji są badania naukowe. Jak widać występuje w tej grze wiele elementów wpływających na powodzenie misji. Aby je kontrolować należy zapoznawać się z bieżącymi raportami doradców. UTOPIA jest wspaniałą grą strategiczną, która doskonały scenariusz łączy z dobrym wykonaniem. Grafika jest na wysokim poziomie. Ikony są bardzo przejrzyste i czytelne. Doskonale rozwiązano sterowanie (przy pomocy ikon). Niesamowita jest atmosfera gry. Uważam, że jest to wspaniała gra łącząca zalety gier strategicznych i ekonomicznych. Myślę, że niejednemu wielbicielowi tego typu gier dostarczy ona zajęcia na długi okres czasu.

Ocena	super!					
Grafika	*	*	*	*	*	*
Dźwięk	*	*	*	*	*	*
Pomysł	*	*	*	*	*	
Motywacja	*	*	*	*	*	*
OGÓLNIE	*	*	*	*	*	s



MANIC MINER

SOFTWARE PROJECTS

Wielkim sukcesem cieszyła się gra HEAD OVER HEELS, która była niczym więcej, jak tylko przeniesieniem hitu z 8-bitowego komputera na ekran Amigi. Przykładem podobnego działania programistów jest gra MANIC MINER. Jest to typowa platformówka polegająca na zbieraniu kluczyków i szukaniu wyjścia z kolejnych komnat. Wymaga od nas perfekcyjnego wprost opanowania joysticka oraz myślenia, gdyż nie wystarczy bezmyślnie skakać i chodzić po komnatach, aby ukończyć tę grę. Do starych plansz dołożono wiele nowych, dlatego nawet Ci, którzy opanowali przejście gry na Spectrum będą musieli solidnie popracować, aby sprostać wymaganiom jej "młodszej siostry". Gra ma niezłą grafikę (stare plansze zostały przeniesione pixel po pixelu) oraz fatalną wręcz oprawę muzyczną (chyba nawet gorszą od swojego pierwowzoru z roku 1983 !!). Całość jest jednak wspaniałym połączeniem gry platformowej z ciekawą grafiką i dużą porcją dobrej zabawy.

Ocena	dobra plus					
Grafika	*	*	*	*	*	
Dźwięk	*	*	*	*	*	
Pomysł	*					
Motywacja	*	*	*	*	*	
OGÓLNIE	*	*	*	*		



CRUISE FOR A CORPSE

DELFIN SOFTWARE

Pod wpływem produktów firmy DELPHIN SOFTWARE pt. FUTURE WARS i OPERATION STEALTH wielu z nas spędziło dużo bezsennych nocy i nie raz poszło do szkoły z mętnym wzrokiem rozmyślając o tym jak wykorzystać maszynę do pisania, czy też tą malutką flagę znalezioną w łazience. Obecnie do naszych rąk trafił najnowszy produkt tej firmy gra CRUISE FOR A CORPS. W stosunku do poprzedniczek CFAC wyróżnia się doskonałą grafiką, bardzo realistyczną perspektywą, płynną i staranną animacją oraz świetnie dopasowaną muzyką. Akcja gry toczy się na pokładzie luksusowego jachtu twojego przyjaciela. Zaprosił on Cię na wspólny rejs, abyście mogli pogawędzić o "starych dobrych czasach". Niestety, gdy przybyłeś na jacht dowiadujesz się, że Twój towarzysz został zamordowany. Postanawiasz samotnie wykryć sprawcę tej okrutnej zbrodni. Rozpoczynasz mozolne i niebezpieczne śledztwo. Penetrujesz poszczególne kabiny oraz wielokrotnie przesłuchujesz załogę i pasażerów. Życzę powodzenia i sukcesów w walce z mordercą przyjaciela.

Ocena	super!					
Grafika	*	*	*	*	*	*
Dźwięk	*	*	*	*	*	
Pomysł	*	*	*	*	*	*
Motywacja	*	*	*	*	*	*
OGÓLNIE	*	*	*	*	*	ξ

Opracował *Jacek Ryć.*

TRIKI

Dzisiaj kolejna porcja trików nadesłanych przez Grzegorza Jaworksiego z Wrocławia. Dysk Public Domain w drodze.

AFTER THE WAR

Kod do drugiej części gry to 101069

INDIANA JONES III ACTION

Nieśmiertelność uzyskuje się po w pisaniu na listę rekordów słowa SILLYMAN.

NARCO POLICE

Jeżeli Cię zabiją to nie kasuj gry, tylko wpisz kod ABORT, bez zmiany dysku i grasz od początku. Zmiana poziomu FUNKCYJNY + RETURN.

NITRO

Podczas wpisywania imienia wpisz MAY. Otrzymasz 50 sztuk pieniędzy i 5000 galonów paliwa.

ROBOCOP II

W czasie dema wpisz SERIALINTERFACE. Podczas gry F9 podnosi energię, F10 następny etap gry.

ELF

Wpisz CHEAT na jednym ze screenów tytułowych. Następnie w czasie gry wpisz CHOROPOO. Otrzymasz wtedy 99% enrgii i jej poziom powróci do stanu początkowego.

SUPERCARS 2

Wpisz nazwisko gracza pierwszego jako WONDERLAND, drugiego jako THE SEER i będziesz wtedy uzbrojony jak Arnold Scwarzeneger.

GHOSTBUSTERS 2

Na obrazku tytułowym wciśnij ALT, CONTROL, S i U. Da Ci to nieskończoną ilość energi na pierwszych dwóch poziomach.

Opracował *Ryszard Kowalski*

COMMODORE AMIGA 500

Wspaniała grafika

Autentyczna animacja trójwymiarowa

Dla szkół, do pracy, dla rozrywki



